

Vehículo Definido por Software

Un Enfoque de Lo Digital Primero para Crear
Experiencias de Próxima Generación

by

Dirk Slama, Achim Nonnenmacher y
Thomas Irawan

Traducción y edición: Aldo Núñez Tovar

To download:

https://github.com/lizard20/AUTOSAR/blob/main/sdv_Slama_Nonnenmacher_Irawan.pdf

Prefacio y Agradecimientos

Esta publicación es el resultado del intenso trabajo que realizamos para la iniciativa digital.auto, que reúne a fabricantes de equipos originales (OEM) y proveedores, miembros de consorcios industriales y entusiastas del código abierto para ayudar a nuestra industria a hacer realidad el vehículo definido por software (SDV).

La iniciativa digital.auto está abierta a todos y es independiente del proveedor y la tecnología. Buscamos impulsar la adopción de SDV centrándonos en casos de uso y en cómo implementarlos utilizando tecnologías y metodologías de vanguardia. Si esta iniciativa le interesa, consulte el [Capítulo 6, «Próximos Pasos»](#).

Esta publicación no habría sido posible sin el apoyo y la retroalimentación de varias personas. Queremos expresar un agradecimiento especial a los colaboradores del curso original SDV 101:

- Achim Henkel, Director, Robert Bosch Group
- Alex Oyler, Director, SBD Automotive
- Alexander Djordjevic, Director de Gestión de Soluciones en RideCare, Robert Bosch GmbH
- Andre Marquis, Profesor Titular/Profesor, UC Berkeley, Haas School of Business, Programa de Emprendimiento
- Ansgar Lindwedel, Director de Desarrollo del Ecosistema SDV, ETAS GmbH
- Daniel Riexinger, Gerente Sénior de IA y Modelos de Negocio Basados en Datos, Mercedes-Benz AG
- Dominik Rose, Vicepresidente de Gestión de Producto y Estrategia de Plataforma, LeanIX GmbH
- Frédéric Merceron, Director de Soluciones de Transporte y Movilidad, Dassault Systèmes
- Georg Hansbauer, Director Ejecutivo, Testbirds GmbH
- Jacek Marczyk, Director Ejecutivo, Ontonix
- Jann Kirchhoff, Gerente de Éxito de Producto, BMW
- Pei Shen, Director General de Estrategia para Tencent Intelligent Mobility, Tencent Inc.
- Sasha Milinkovic, Gerente, mm1 Consulting GmbH
- Sebastian Werner, Director de Software Automotriz, Kearney & BinaryCore
- Sven Kappel, Vicepresidente de Vehículos Definidos por Software, ETAS GmbH
- Tom Acland, Director Ejecutivo, Dassault Systèmes 3DEXCITE

Asimismo, agradecemos a la alta dirección de Bosch su apoyo a digital.auto, en particular a Tanja Rückert (CDO, Grupo Bosch) y a Andreas Dempf. (CSO/CMO, Bosch Mobility) y Mathias Pillin (CTO, Bosch Mobility), así como a los directivos del Instituto Ferdinand-Steinbeis: Prof. Heiner Lasi y Michael Köhnlein. Agradecemos a Chris Cheng y al equipo del Laboratorio de AIoT del Instituto Ferdinand-Steinbeis por su constante apoyo a digital.auto y a diversos casos de uso (esta investigación fue financiada por Dieter Schwarz Stiftung); a Christian Seiler, de Mercedes-Benz, por sus intensos debates sobre vehículos definidos por software y la gestión de variantes; a Damian Dyrbusch, de Bosch, por su visión estratégica; a Can Yasin Gümüş, de Bosch, por sus aportaciones sobre SDV y ADAS; a Hoàng Phan Thanh, Nhân Lu'ông Nguyen, Phong Pham Tuan y Tam Thai Hoang Minh, de Bosch Vietnam, por su trabajo en el entorno digital.auto. Oliver Kust por su guía experta en seguridad automotriz; Marco Wagner por los debates en torno a SDV y la abstracción del vehículo; Imran Abdul Rahiman de ETAS (una empresa de Bosch) por los intensos debates sobre la Arquitectura de Sistemas Abiertos Automotrices (AUTOSAR); Louis Jackman por la versión inicial del área de juegos digital.auto; Mohan B. V. y Babu Niranjan de Bosch Global Software Technologies por su orientación sobre la estructura general de este libro; Grace Lee de Landing.AI por mostrarnos cómo se puede democratizar la IA; Boris Scholl de Microsoft por sus aportaciones sobre Cloud Native; Francesco Salamida y a Stefano Marzani de AWS por las conversaciones sobre innovaciones en el ámbito de los SDV; a Wieland Holfelder de Google por sus aportaciones al entorno digital.auto; y, por último, a nuestras familias por su apoyo incondicional mientras dedicábamos horas extras a este proyecto. - ¡Gracias!

— Dr. Dirk Slama
Vicepresidente de Bosch
Profesor del Instituto Ferdinand-Steinbeis

— Dr. Achim Nonnenmacher
Gerente Sénior de Innovaciones SDV
en ETAS (una empresa de Bosch)

— Dr. Thomas Irawan
Director General de ETAS (una empresa de Bosch)

Berlín/Stuttgart, agosto de 2023

A continuación, se presentan algunos términos clave que se encuentran en este folleto:

- API: Interfaz de Programación de Aplicaciones
- AUTOSAR: Arquitectura de Sistema Abierto Automotriz
- CAN: Red de Área del Controlador
- COVESA: Alianza de Sistemas de Vehículos Conectados
- CI/CD: Integración y entrega continuas
- E/E: Arquitectura eléctrica/electrónica del vehículo
- ECU: Unidad de Control Electrónico
- OEM: Fabricante de Equipo Original
- SOA: Arquitectura Orientada a Servicios
- SOP: Inicio de la Producción
- QoS: Calidad de Servicio
- vECU: Unidad de Control Electrónico Virtual
- VSS: Especificación de Señales del Vehículo (de COVESA)

Índice general

Preface	II
1 Introducción	1
2 Lo que la Industria Automotriz Puede Aprender de Otras Industrias	12
3 Sistema operativo (SO) vehicular y tecnologías facilitadoras	17
4 Gestión del Flujo de Valor para el SDV	28
5 #digitalfirst: Una Nueva Forma de Trabajar	33
6 Siguiendo pasos	38

1

Introducción

En el pasado, la industria automotriz era un testimonio de la potencia de los motores de combustión y del prestigio de poseer un auto con la mayor cantidad de tubos de escape. Hoy, este paradigma anticuado está experimentando una transformación radical. Cuatro innovaciones importantes —electrificación, automatización, movilidad compartida y movilidad conectada— se están produciendo simultáneamente, provocando cambios drásticos en el panorama automovilístico. Amplificando aún más esta complejidad están los recién llegados a la industria, que no solo fabrican autos de alta calidad, sino que se centran específicamente en el consumidor digital con sus “smartphones sobre ruedas”: vehículos equipados con grandes pantallas interactivas, conectividad fluida y actualizaciones frecuentes de software.

Al mismo tiempo, los clientes expresan su insatisfacción porque sus vehículos carecen de las funciones y la facilidad de uso habituales en los teléfonos inteligentes. Muchos se preguntan: ¿por qué mi coche de \$50,000 no puede realizar las mismas funciones que mi teléfono inteligente de \$300?

De esta frustración surgió la idea de un vehículo definido por software (SDV), un coche totalmente programable. Se pueden desarrollar e implementar nuevas funciones en cuestión de meses, no años, y cuenta con capacidad computacional adicional para futuras actualizaciones que pueden entregarse de forma inalámbrica. Este coche mejora constantemente gracias a la incorporación continua de nuevas y valiosas funciones y actualizaciones de software a lo largo de su vida útil.

Sin embargo, la industria automotriz enfrenta obstáculos que superar si desea alcanzar este objetivo rápidamente. La complejidad del software se está descontrolando, con un volumen de código que alcanza aproximadamente 300 millones de líneas con la conducción autónoma de nivel 5, mientras que los desarrolladores se aferran a las metodologías de desarrollo de software de antaño. Además, el hardware y el software siguen estando estrechamente integrados y se lanzan simultáneamente, lo que significa que los cambios importantes solo ocurren aproximadamente cada siete años.

Consideremos la industria de la telefonía móvil, donde marcas líderes como Nokia y BlackBerry fueron rápidamente superadas por los teléfonos inteligentes. Un destino

similar parece posible para las empresas establecidas de la industria automotriz. Al principio, muchos veteranos de la industria de la telefonía móvil insistieron en que su negocio era fundamentalmente diferente de la industria informática, pero la revolución de los teléfonos inteligentes rápidamente desvaneció esa ilusión.

Este cambio plantea varias preguntas cruciales: ¿Cómo pueden las empresas establecidas de la industria automotriz tener éxito en su continua transformación digital para evitar un destino similar? ¿Qué hace que el cambio sea tan desafiante en esta industria y cómo podemos acelerar significativamente la innovación automotriz? ¿Cómo será el futuro de la generación de ingresos? ¿Seguirán los ingresos provenientes principalmente de las ventas de automóviles, o se convertirán los servicios digitales adicionales en el principal motor de ingresos? Y, fundamentalmente, ¿cómo identificamos las "aplicaciones clave": esas aplicaciones o funciones digitales destacadas que resultan tan esenciales o atractivas que impulsan el éxito de las futuras generaciones de vehículos?

Si bien el SDV es un factor técnico clave, la respuesta a estas preguntas comienza con la adopción de una estrategia "lo digital primero". Esto implica comenzar con la experiencia digital del cliente y avanzar hacia atrás hasta el diseño de la solución, realizando exploraciones y pruebas iniciales para garantizar que los vehículos en circulación se ajusten a las necesidades en constante evolución del consumidor moderno.

Objetivos del SDV

Comencemos nuestra discusión analizando por qué queremos construir un SDV. No se trata de bits y bytes ni de la cantidad de sensores y cámaras, sino de las experiencias de cliente que creamos. Y estas experiencias de cliente no pueden simplemente reconstruir la experiencia del smartphone. Se trata de crear un "hábitat móvil" impulsado por aplicaciones multidominio y fusión de datos, que se entrega y mejora constantemente a una velocidad diez veces superior a la actual.

Smartphones Hábitat sobre Ruedas

Durante la última década, muchos fabricantes de automóviles han buscado replicar aplicaciones exitosas para smartphones en sus vehículos. Sin embargo, en muchos casos, estas aplicaciones integradas en el vehículo no pudieron igualar la calidad de las aplicaciones para smartphones. Además, los consumidores generalmente no desean experiencias redundantes, inconsistencias entre sus ecosistemas digitales ni la molestia de una sincronización de datos engorrosa. Hoy en día, los fabricantes de automóviles deben mantener un equilibrio entre la propiedad de las aplicaciones y los datos, por un lado, y la experiencia del cliente y una mayor integración con los ecosistemas dominantes de smartphones, por otro.

Por eso es importante que el SDV supere el concepto de "smartphone sobre ruedas". En su lugar, debe propiciar un "hábitat sobre ruedas", aprovechando las características específicas del vehículo para ofrecer experiencias multisensoriales que un smartphone jamás podría igualar. Con múltiples pantallas y una red de cientos de sensores y actuadores, el SDV integra ámbitos como el infoentretenimiento, la conducción

autónoma, la carrocería inteligente, el habitáculo y el confort, la energía y los servicios de coche conectado, creando un viaje único.

Los pasajeros se sienten reconocidos al subir a un vehículo personalizado según sus necesidades, un vehículo que rompe con la impersonalidad de los coches tradicionales. Experimentan una reconfortante sensación de seguridad mientras viajan bajo la atenta mirada de sistemas de seguridad de vanguardia. Suben al vehículo con una conciencia ecológica. Reconquistan su tiempo, transformando el viaje en una oportunidad para disfrutar de una experiencia plena, ya sea trabajando, descansando o divirtiéndose. Los pasajeros con discapacidad pueden disfrutar de una nueva libertad y movilidad gracias a la conducción autónoma. Esta tecnología les permite moverse con confianza, manteniendo su independencia intacta. No se trata solo del viaje; se trata de la libertad de explorar, interactuar y vivir la vida sin restricciones.

Los SDV han revolucionado nuestra percepción de la movilidad. Ya no se trata solo de ir del punto A al punto B, sino de hacer que el viaje en sí sea enriquecedor. Gracias a los sistemas avanzados de asistencia al conductor y a la conducción autónoma, estamos adoptando el poder transformador y multisensorial del SDV.

En el SDV, los mundos digital y físico se fusionan, creando experiencias inmersivas e intuitivas. Esta fusión permite al vehículo interpretar y reaccionar a las señales físicas con precisión y sofisticación, mejorando la experiencia general. Por ejemplo, los sensores del vehículo podrían identificar el estado de salud de los niños en el asiento trasero y estimar el posible mareo por movimiento según la curvatura de la carretera y la velocidad. En respuesta a estas sutiles señales, el sistema podría adaptar el estilo de conducción automatizada, ajustar la posición de las ventanillas y regular el aire acondicionado para mejorar la comodidad y minimizar el mareo por movimiento.

Además, las funciones digitales de valor añadido transforman el vehículo en un centro de productividad y entretenimiento. Imagine videoconferencias proyectadas en la ventanilla delantera como hologramas, participar en juegos interactivos con otros conductores durante los atascos o aprender un nuevo idioma según su entorno durante el trayecto, todo ello desde la seguridad y la comodidad de su espacio personal y móvil. Esta perfecta integración de los mundos digital y físico, facilitada por el SDV, garantiza que cada viaje no sea solo un paseo, sino una experiencia enriquecedora que aporta valor mucho más allá de la llegada a su destino.

Aplicaciones multidominio y fusión de datos

Hoy en día, las experiencias con vehículos suelen ocurrir en dominios aislados. Pero el futuro de los SDV promete difuminar la frontera entre el vehículo y el mundo exterior. Las experiencias serán multidominio, donde diversas funciones y sistemas del vehículo se intercomunican e interactúan armoniosamente para enriquecer el viaje en general. Como muestra la Figura 1-1, los SDV representan una transición de funciones dispares a experiencias integradas, donde el vehículo funciona como una entidad unificada en lugar de un conjunto de piezas separadas.

Consideremos el ejemplo de un "modo perro" digital que algunos coches ya incorporan.

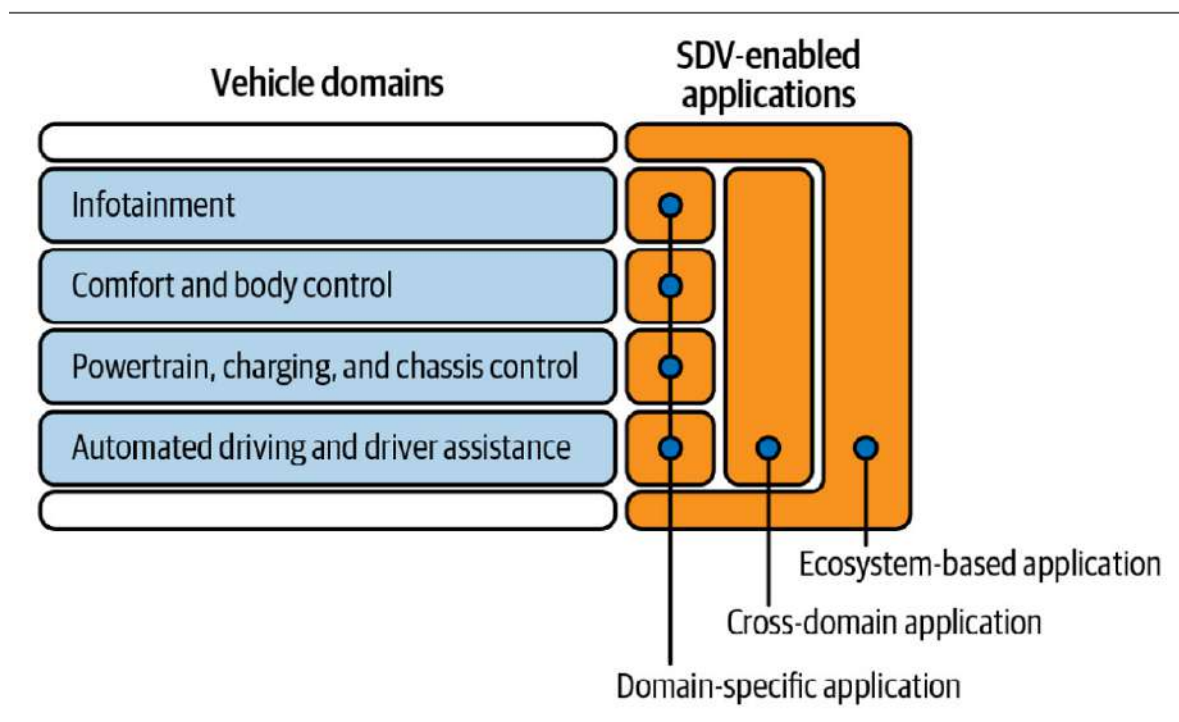


Figura 1.1: Servicios de aplicaciones multidominio

El vehículo vigila a su perro mientras usted va de compras. Dado que a los perros les cuesta enfriarse, un calor intenso en el interior del coche en un día de verano suele ser suficiente para causar lesiones graves o incluso la muerte. Este es un ejemplo perfecto de funcionalidad centrada en el cliente y multidominio. Implica varios sistemas: el aire acondicionado del coche para mantener una temperatura agradable, la pantalla de infoentretenimiento para mostrar un mensaje que avisa a los transeúntes de que no se preocupen, ya que el perro está seguro y cómodo, y el sistema de gestión de la batería para garantizar que el coche tenga suficiente energía. Todos estos dominios están coordinados para garantizar la seguridad y la comodidad del perro.

En este ecosistema conectado, tu coche podría incluso convertirse en una extensión creativa de tu presencia en redes sociales. Con tu permiso, podría capturar una impresionante puesta de sol con sus cámaras integradas de alta calidad durante un recorrido panorámico y proponer una publicación preeditada para tu aprobación. Tu vehículo se convierte en un participante activo de tu vida digital, permitiéndote compartir momentos únicos sin mermar la experiencia de conducción.

Imagina que tu vehículo eléctrico se comunica con una red eléctrica inteligente y programa su carga durante las horas valle. Esta interacción optimiza el consumo energético y maximiza la rentabilidad. Tu coche ya no es solo un consumidor de energía, sino también un participante inteligente en el ecosistema energético.

Las experiencias multidisciplinarias también se extienden al bienestar personal. Imagina que tu dispositivo de fitness te indica que has hecho un entrenamiento intenso. En respuesta, tu coche baja la temperatura del habitáculo, selecciona una iluminación ambiental relajante y reproduce tu lista de reproducción favorita para relajarte. Al

integrarse a la perfección con tus dispositivos digitales, tu coche mejora tu recuperación y comodidad después del entrenamiento. Esta fusión de experiencias digitales y físicas convierte al coche en una extensión de tu vida digital, conectándote a la perfección con el mundo que te rodea.

Acelerar la innovación diez veces

Para lograr el desarrollo centrado en el cliente y la mejora continua necesarios para alcanzar los objetivos de usabilidad descritos anteriormente, el desarrollo de SDV debe acelerarse significativamente, acortando así el plazo de comercialización de las nuevas funciones. La Figura 1-2 muestra la transformación que debe producirse.


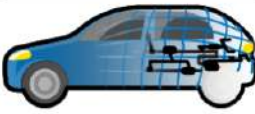
	 Today's typical vehicle	 Software-defined vehicle
Development cost	\$\$\$\$	\$
First prototype	Months	Hours
Time to market	3 years	Weeks to months
Approach	First-time right at HW SOP	MVP and continuous improvement
Scalability	Every car model needs different code	Same code for all car models

Figura 1.2: "Necesidad de velocidad" en el desarrollo de software automotriz.

El objetivo debería ser una mejora diez veces superior a la forma tradicional de ofrecer nuevas funciones. Los costes de desarrollo deben reducirse significativamente. Los prototipos iniciales deben estar disponibles en horas, no en meses. El tiempo de comercialización debe reducirse de años a unas pocas semanas.

Y todo esto no se aplica solo a las nuevas funciones: debemos ser capaces de monitorizar constantemente cómo los clientes utilizan las funciones existentes, aprender a mejorarlas y luego optimizarlas; todo ello, una vez más, diez veces más rápido de lo que hoy se hace.

Impedimentos: ¿Por qué es diferente el desarrollo de software automotriz?

Para alcanzar nuestros objetivos para el SDV, debemos comprender los impedimentos que debemos superar. ¿Por qué el desarrollo de software automotriz es diferente al desarrollo en otros dominios (por ejemplo, los teléfonos inteligentes)? Sin duda, la metáfora de un "smartphone sobre ruedas" es accesible e ilustra la naturaleza evolutiva de los SDV. Con el auge de las pantallas táctiles, la conectividad robusta y la gran cantidad de funciones basadas en aplicaciones en nuestros automóviles, establecer un paralelo con nuestros dispositivos inteligentes portátiles parece acertado. Sin embargo, el encanto simplista de esta metáfora puede ser engañoso, ya que no tiene en cuenta las complejidades y los desafíos únicos del sector automotriz.

El siguiente análisis ofrece un análisis profundo de los principales impedimentos para la adopción generalizada del SDV en la industria automotriz, incluyendo la complejidad y la heterogeneidad, la seguridad funcional, el "choque de dos mundos" y la necesidad de una transformación organizacional.

Complejidad y heterogeneidad

Una diferencia clave entre la industria automotriz y la de los teléfonos inteligentes reside en sus niveles de complejidad. Durante la última década, la industria de los teléfonos inteligentes ha logrado abordar la complejidad técnica mediante la estandarización y la abstracción. Los teléfonos inteligentes actuales están altamente integrados con múltiples capas de abstracciones e interfaces. La industria automotriz, por otro lado, aún se ve afectada por altos niveles de complejidad y heterogeneidad.

Además, los altos niveles de abstracción han permitido a la industria de los teléfonos inteligentes encapsular la mayor parte de la complejidad del hardware, lo que le permite gestionar la complejidad a nivel de software. Por otro lado, los vehículos actuales siguen siendo complejos sistemas de sistemas. Cada subsistema de un automóvil, desde los frenos hasta el sistema de propulsión, es una entidad compleja, suministrada por un proveedor diferente e integrada con una arquitectura de software única. El nivel de complejidad y la necesidad de una interoperabilidad fluida entre sistemas supera con creces lo que vemos en los teléfonos inteligentes actuales.

Otra diferencia clave es que la industria automotriz maneja muchas más variantes de producto. Muchos autos se configuran según las preferencias individuales de los clientes, expresadas durante el proceso de venta. Las diferencias regionales en las preferencias de los clientes y los requisitos regulatorios dan lugar a un gran número de variantes de producto. Y los diferentes requisitos funcionales (como furgonetas familiares, descapotables o camiones) dan lugar a una gran variedad de tipos, modelos, ediciones y variantes de vehículos. Por lo tanto, la producción por modelo de vehículo suele ser mucho menor que por modelo de smartphone.

Seguridad funcional

La comparación de “teléfono inteligente sobre ruedas” también se queda corto cuando consideramos los requisitos de seguridad funcional automotriz. A diferencia de los teléfonos inteligentes, los vehículos están sujetos a estándares de seguridad estrictos, como ISO 26262. Este estándar se ocupa de la seguridad funcional de los sistemas eléctricos y electrónicos dentro de los vehículos y es fundamental para el concepto de un SDV.

El estándar ISO 26262 no solo ofrece un conjunto de reglas, sino que incorpora una filosofía basada en el riesgo que evalúa los peligros potenciales y requiere técnicas de diseño apropiadas para mitigarlas. Los riesgos se clasifican según tres factores: gravedad (el daño potencial que puede ocurrir), exposición (la frecuencia probable de la situación de riesgo), y controlabilidad (la capacidad del conductor para evitar el peligro). Al evaluar estos factores, el estándar equilibra efectivamente la innovación y la seguridad en el diseño automotriz.

Para cuantificar el riesgo, el estándar emplea un marco conocido como niveles de integridad de seguridad automotriz (ASIL), ilustrado en la Figura 1-3, que clasifica los eventos peligrosos que podrían resultar de un mal funcionamiento basado en su nivel de gravedad, exposición y capacidad de control. Los niveles de riesgo van desde ASIL A, el nivel más bajo, hasta ASIL D, el nivel más alto. ISO 26262 define los requisitos y medidas de seguridad que se aplicarán en cada ASIL.

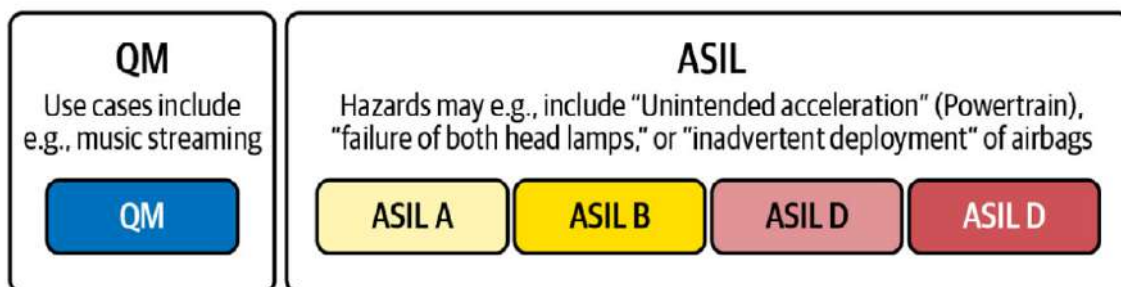


Figura 1.3: Niveles de seguridad automotriz

Además, se asigna una clase llamada QM (calidad administrada) a sistemas que, después de un análisis exhaustivo, solo requieren procesos de gestión de calidad estándar debido a su menor impacto potencial en la seguridad. Por ejemplo, una función relevante para la seguridad como un sistema de frenado se le asignaría una calificación ASIL alta, y se le asignaría un sistema de entretenimiento en el automóvil una calificación de QM. El software que controla los frenos necesitaría adherirse a estándares de seguridad más rigurosos que el sistema de entretenimiento. El objetivo general en el diseño del vehículo es la mitigación de riesgos, un proceso que busca reducir los peligros potenciales.

Por ejemplo, el riesgo de una señal de velocidad falsa debido a un sensor defectuoso puede mitigarse introduciendo un sensor redundante. Este sensor adicional puede validar las señales, minimizando así las posibilidades de error y mejorando la seguridad general del vehículo.

Por lo tanto, si bien la analogía del “teléfono inteligente sobre ruedas” retrata sucintamente el papel emergente del software en los vehículos, no captura por completo los estrictos estándares de seguridad y las rigurosas estrategias de mitigación de riesgos empleadas en la industria automotriz. El SDV es más que un simple dispositivo móvil; Es un conjunto de sofisticado de sistemas que prioriza la seguridad tanto como la funcionalidad y la conveniencia.

Choque de dos mundos

La ingeniería automotriz se ha centrado tradicionalmente en la funcionalidad física, desde características electrónicas tempranas (como bolsas de aire, estabilización de vehículos y sistemas de frenado) hasta sistemas modernos de asistencia del conductor o incluso conducción automatizada. La nueva experiencia digital es impulsada por características habilitadas por software, y el enfoque es mejorar la experiencia de usar el vehículo o incluso proporcionar características digitales de valor agregado como seguro basado en el uso o el pago automatizado de las tarifas de estacionamiento. La Figura 1-4 muestra algunas de estas diferencias.

El mundo tradicional de las características de vehículos físicos de ingeniería se caracteriza por un complejo sistema de sistemas, conformando con la seguridad funcional, la seguridad y los requisitos en tiempo real, y la homologación, un proceso para obtener la certificación aprobada por el gobierno para la preparación del mercado de productos para garantizar que se cumplan los estándares de seguridad y ambientales. Por el contrario, la nueva experiencia de vehículo digital se basa en las mejores prácticas del sector tecnológico, incluidos los métodos ágiles y el desarrollo tipo-nube.

Los vehículos modernos son una fusión de estos mundos; deben fusionar la confiabilidad de la ingeniería tradicional con la agilidad del desarrollo de software moderno (ver Figura 1-5). Esto presenta a la industria automotriz con un desafío significativo: los OEM que mejor pueden abordar este “choque de dos mundos” tienen más probabilidades de tener éxito.

Transformación organizacional

Podemos observar que durante la última década, los OEM titulares han sufrido una transformación organizacional compleja y a gran escala para ofrecer características de automóviles físicos y digitales. Es vital que los OEM permitan a sus organizaciones combinar los mundos de la ingeniería tradicional y el desarrollo de software moderno y hacer que trabajen juntos en relativa armonía. Pero esto significa que deben crear organizaciones con suborganizaciones que puedan moverse a diferentes velocidades y trabajar con diferentes culturas, métodos y herramientas. No es trivial crear plataformas y arquitecturas de vehículos que sean tan modulares que los componentes con diferentes requisitos (por ejemplo, seguridad funcional) pueden asignarse a las unidades organizativas que son los que mejor encajan.

Esto no se detiene con la organización interna de la compañía. Tradicionalmente, muchos OEM compraron diferentes subsistemas de diferentes proveedores, generalmente

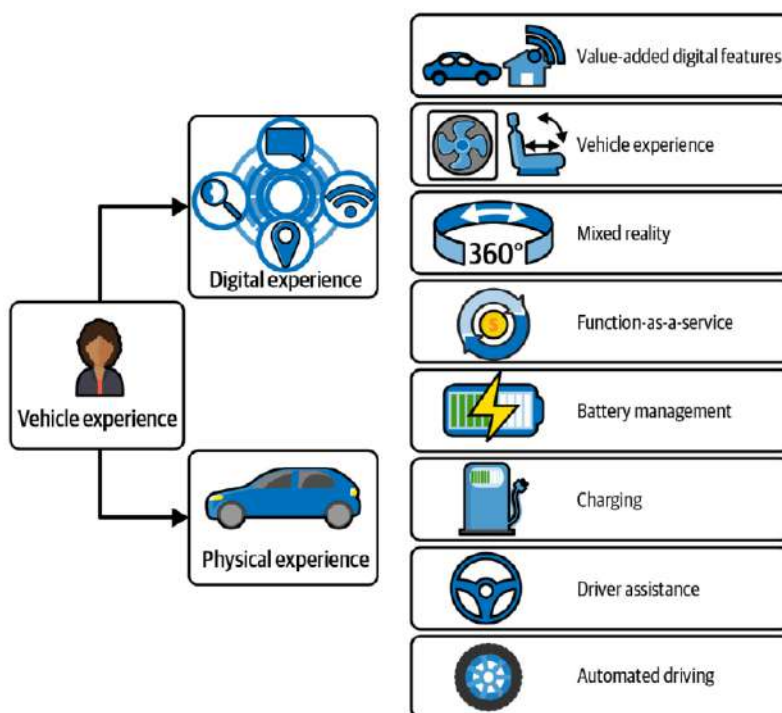


Figura 1.4: Experiencia digital versus experiencia física

combinando hardware y software para cada subsistema. Sin embargo, con los SDV, el desacoplamiento del hardware y el software debe considerarse con respecto al abastecimiento. Las cadenas de suministro establecidas están experimentando transformaciones significativas. Los roles de los proveedores existentes están cambiando, y los nuevos proveedores están entrando en el campo de juego.

Repensar el Ciclo de Vida del Vehículo: Lo Digital Primero

¿Qué podemos aprender de la discusión de los impedimentos que se pueden aplicar al ciclo de vida del vehículo? Históricamente, el ciclo de vida de un vehículo se definió por la producción y el despliegue simultáneos de hardware y software estrechamente acoplados. Una vez que el vehículo estaba en manos del consumidor, sus características permanecieron esencialmente inalteradas hasta el fin de su vida. Sin embargo, un paradigma SDV permite el desacoplamiento de fechas de lanzamiento de hardware y software, un requisito previo para el enfoque lo digital primero, que pone el diseño y la validación virtual de la experiencia del vehículo digital al comienzo del ciclo de vida.

Lo digital primero significa que las nuevas ideas para la experiencia del vehículo se exploran inicialmente en entornos virtuales, recibiendo comentarios tempranos del usuario mucho antes de que se desarrolle un hardware personalizado o incluso se proporcione un vehículo de prueba física. Lo digital primero significa que los principios del pensamiento de diseño y la startup lean, que se originó en la cultura de Internet, ahora se pueden aplicar en el mundo físico del desarrollo automotriz.

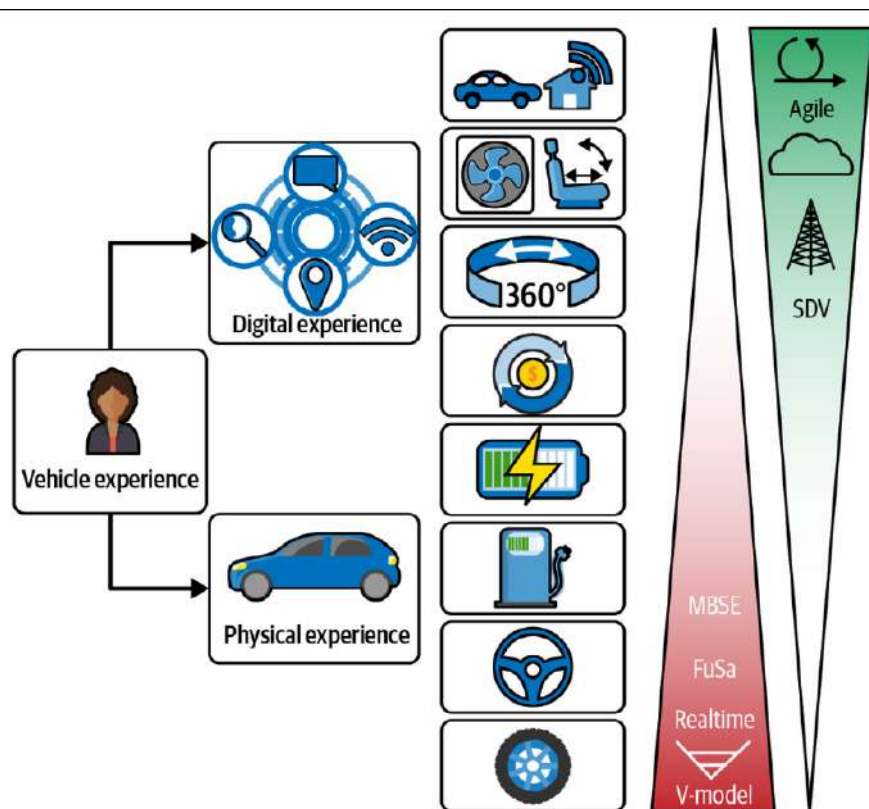


Figura 1.5: Choque de dos mundos

En el modelo SDV, se generan nuevas ideas con enfoques como el pensamiento de diseño, donde las características se exploran virtualmente y se prueban con la retroalimentación temprana de los clientes, lo que facilita una forma más rápida y rentable de encontrar la adecuación producto-mercado. La adecuación producto-mercado se refiere al estado en el que un producto satisface las necesidades y deseos específicos de su mercado objetivo, en este caso, para nuevas aplicaciones y servicios para vehículos. La capacidad de explorar, innovar y adaptarse rápidamente es fundamental en la nueva era automotriz.

Mucho de esto se relaciona con nuestra conversación sobre los smartphones. Así como el lanzamiento del iPhone abrió las puertas a aplicaciones inimaginables —identificar especies de plantas, optimizar la calidad del sueño, estudiar para obtener el carnet de conducir—, la llegada del SDV nos impulsa hacia un futuro de posibilidades inimaginables. Las aplicaciones y experiencias que hacen posibles estos hábitats inteligentes sobre ruedas permanecen en gran parte sin explotar, invitando a las mentes creativas a explorarlas.

No podemos predecir con precisión cuáles podrían ser estas aplicaciones revolucionarias. Es probable que la "aplicación estrella" del SDV no sea solo un equivalente a *Angry Birds* para el coche, sino algo que aproveche el potencial único de este nuevo entorno móvil.

Pero podemos afirmar con seguridad lo siguiente: para descubrir estas aplicaciones y experiencias innovadoras, es necesario experimentar e innovar a gran escala. Esto requiere un desarrollo mucho más rápido, como ya comentamos. Sin embargo, también

requiere un nuevo enfoque para validar nuevas ideas, especialmente desde el punto de vista de la conveniencia y la usabilidad. Construir vehículos de prueba físicos es muy costoso y requiere mucho tiempo. Por ello, la industria automotriz busca formas de virtualizar las pruebas de usabilidad y aceptación. Por ejemplo, el entorno de pruebas digital.auto proporciona un entorno que permite probar nuevas ideas para las funciones de los vehículos digitales en un entorno de nube pura y evaluarlas con las API de vehículos reales. La aparición de la computación espacial y la realidad virtual acelerará las pruebas de experiencia de usuario -UX- virtualizadas. Es importante que estas pruebas no se limiten al diseño físico del vehículo. La capacidad de probar la experiencia del vehículo, tal como la habilita SDV, en entornos virtuales facilitará significativamente las pruebas de usuario de desplazamiento a la izquierda. Esto, a su vez, ayudará a garantizar que las inversiones se prioricen según la demanda del mercado, y que la experiencia del vehículo resultante genere productos populares y altos niveles de lealtad del cliente. La figura 1-6 resume el ciclo de vida del vehículo digital.

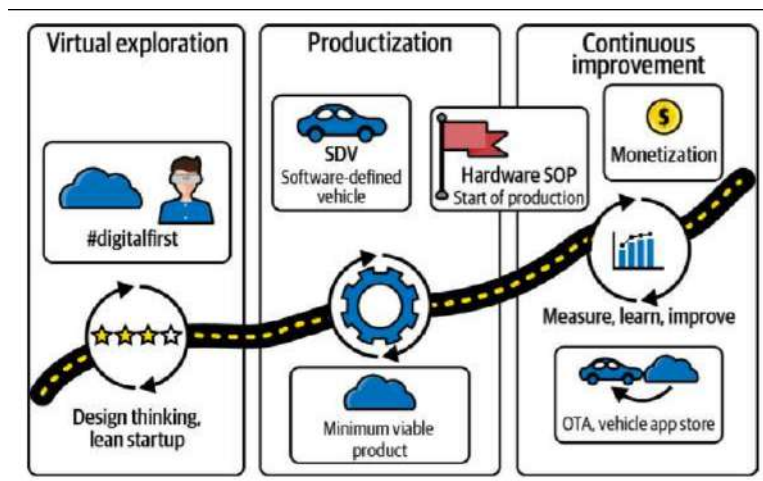


Figura 1.6: Ciclo de vida de un vehículo Lo Digital Primero

A diferencia de antes, cuando los fabricantes de automóviles tenían un conocimiento limitado del uso real de ciertas funciones, los SDV pueden, con el consentimiento del conductor, medir y evaluar el uso real de las aplicaciones del vehículo y sus funciones. Este enfoque basado en datos fomenta un ciclo de aprendizaje rápido, lo que facilita innovaciones adaptadas al uso y las preferencias del cliente. **El capítulo 5, «#digitalfirst: Una Nueva Forma de Trabajar»,** ofrece más detalles sobre cómo aplicar todo esto en la práctica.

2

Lo que la Industria Automotriz Puede Aprender de Otras Industrias

En el Capítulo 1, exploramos la visión del mundo del software automotriz moderno, profundizando en el concepto de un "hábitat móvil para smartphones" y abordando los desafíos y oportunidades únicos que presenta. Demostramos la necesidad de innovar a un ritmo acelerado y las complejidades de garantizar la seguridad funcional en este entorno híbrido.

Ahora nos centraremos en los conceptos y tecnologías fundamentales que sustentan esta visión. Analizaremos qué impulsó el éxito de la revolución de los smartphones e internet y cómo podemos aplicar estas lecciones al SDV.

Aprendiendo de los Usuarios de Smartphones: Estandarización, Abstracción de Hardware y Tiendas de Aplicaciones

La incursión de Nokia en el mundo móvil ofrece una lección importante para el SDV. Para 2009, Nokia había creado un laberinto con 57 versiones diferentes e incompatibles de **su sistema operativo**. Las consecuencias fueron devastadoras. Para los desarrolladores, crear aplicaciones se convirtió en una tarea abrumadora debido a la gran fragmentación. Para los usuarios, el limitado ecosistema de aplicaciones resultante hizo que la plataforma de Nokia perdiera su atractivo.

Esta historia refleja un desafío actual en el mundo automotriz. Hoy en día, casi todos los modelos de autos, incluso los de un solo fabricante, emplean componentes de hardware y software personalizados de diversos proveedores. El resultado: una fragmentación extrema combinada con marcos de programación monolíticos, donde crear una "aplicación para vehículos" compatible con múltiples modelos del mismo fabricante parece casi imposible.

Sin embargo, la industria de los smartphones ofrece un plan para superar esta

fragmentación. Su solución fue multifacética:

- Las interfaces de programación de aplicaciones (API) estandarizadas proporcionan una interfaz consistente para los desarrolladores, independientemente de las diferencias subyacentes del sistema.
- Un lenguaje de programación y un entorno de desarrollo de aplicaciones fáciles de usar garantizan que los desarrolladores experimenten una curva de aprendizaje fluida y un entorno encapsulado para sus aplicaciones.
- Una tienda de aplicaciones actúa como un centro de distribución centralizado, permitiendo a los desarrolladores externos poner sus aplicaciones a disposición de los usuarios finales sin problemas.

Entonces, ¿qué debería aprender la industria de SDV de esto? Hay al menos tres lecciones:

Entonces, ¿qué debería aprender la industria de SDV de esto? Hay al menos tres lecciones:

1. *API unificadas*. Un conjunto de API estandarizadas para vehículos simplificaría enormemente el proceso de creación de software. Al garantizar que estas API tengan una fragmentación mínima, los desarrolladores podrían escribir software una sola vez y que funcione en múltiples modelos de vehículos. Basándonos en las tendencias del sector, creemos que, a largo plazo, habrá una tendencia natural hacia unos pocos conjuntos de API dominantes, lo que minimizará la fragmentación en toda la industria. Dentro de cada OEM, prevemos un enfoque de API más unificado, lo que generará consistencia en sus líneas de producto individuales.
2. *Capa de abstracción de hardware (HAL)*. Actúa como puente entre las aplicaciones de software y la multitud de variaciones de hardware del vehículo. Garantiza que el software pueda ejecutarse independientemente de las diferencias de hardware subyacentes, lo que añade consistencia y previsibilidad.
3. *Pila de software de soporte (sistema operativo del vehículo)*. Una pila de software robusta, en armonía con las API estandarizadas y HAL, garantiza que el software pueda interactuar fluidamente con los componentes del vehículo, facilitando la introducción y adopción de innovaciones basadas en software.

La Figura 2-1 ofrece una visión general de cómo las mejores prácticas de la industria de los teléfonos inteligentes pueden transferirse a la industria automotriz. Esta visión general considera que el sector automotriz incluye áreas con requisitos de procesamiento en tiempo real y seguridad funcional. Un obstáculo clave es la necesidad de los OEM de diferenciarse. Los grandes OEM, en particular, querrán mantener gran parte de su desarrollo como propiedad exclusiva, mientras que los OEM más pequeños podrían tener un mayor interés en la estandarización y la reutilización en toda la industria. La solución en este caso puede ser estandarizar los marcos y permitir la diferenciación en cuanto al contenido. Por ejemplo, un OEM podría querer estandarizar y abrir entre el 70% y el 80% de las API que cubren funciones de productos básicos. Utilizando el mismo marco técnico, este OEM podría gestionar entre el 20% y el 30% de las API que

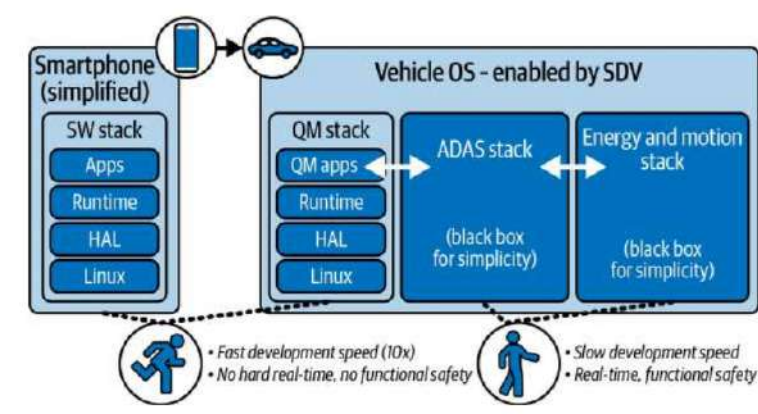


Figura 2.1: Aplicación de las mejores prácticas de los teléfonos inteligentes al desarrollo automotriz

desea diferenciar de forma propietaria.

Mohan B.V., Director de Estrategia Tecnológica de Mobility Next en Bosch BGSW, afirma:

Estamos convencidos de que, en el futuro, muchas funciones obligatorias serán compatibles con todos los fabricantes de equipos originales (OEM). ¿Tiene sentido desarrollar API específicas para cada fabricante? La interoperabilidad fluida de datos entre el vehículo, el proveedor de servicios y el conductor es clave, y una API armonizada lo permitirá; esto es lo que hemos aprendido de la industria de los smartphones.

Can Yasin Gümüş, Director Sénior de Proyectos de Gestión de la Innovación en Bosch XC, añade:

La definición de una API estandarizada abre el mercado a una estrategia comunitaria. Por ejemplo, resolver la conducción automatizada y la asistencia al conductor supone grandes retos. Una de las razones son los complejos problemas regionales que enfrentan los sistemas. Al facilitar el acceso al espacio de soluciones, los desarrolladores de todo el mundo pueden contribuir a la solución de este tipo de problemas. SDV abre un mercado que permite la democratización del desarrollo de la conducción automatizada y la asistencia al conductor.

Aprendiendo de los expertos en internet: Código abierto y desarrollo nativo de la nube

Incluso antes de la aparición de los teléfonos inteligentes, internet transformó el mundo al revolucionar la comunicación, el acceso a la información, el comercio y la interacción social. Entre los factores clave de este cambio se incluyen las tecnologías nativas de la nube, la democratización del conocimiento y el desarrollo de código abierto. El término *nativo de la nube* se refiere a un enfoque de arquitectura y desarrollo de software que aprovecha los principios y servicios de la computación en la nube para permitir una

rápida adaptación a las demandas del mercado, la escalabilidad a nivel de internet y la resiliencia. Hoy en día, muchos de los elementos de software necesarios para el desarrollo nativo de la nube cuentan con el respaldo de una gran comunidad global de código abierto.

Como se muestra en la Figura 2-2, los elementos clave de la nube nativa incluyen microservicios y contenedorización de API, integración y entrega continuas (CI/CD), y desarrollo y operaciones de sistemas de TI (DevOps).

Microservicios y API

Los microservicios son componentes de software que encapsulan sus datos y lógica de negocio, y los ponen a disposición mediante API bien definidas. Dado que las arquitecturas de microservicios están débilmente acopladas, son ideales para dar soporte a equipos interorganizacionales que trabajan en múltiples microservicios y evolucionan a diferentes velocidades.

Contenedorización

Los contenedores proporcionan el entorno de ejecución nativo de la nube para microservicios. Los contenedores proporcionan virtualización a nivel de aplicación, lo cual es más ligero que la virtualización de un sistema operativo completo. Los contenedores suelen implementarse en múltiples nodos de red en la nube, lo que proporciona escalabilidad y resiliencia a los microservicios que se ejecutan en ellos. También proporcionan niveles adicionales de aislamiento, seguridad y gestión de sistemas.

Entrega continua y DevOps

La entrega continua y DevOps son prácticas modernas de desarrollo de software que garantizan que las complejidades, la dinámica y las incertidumbres de los mercados actuales puedan ser soportadas por cambios de código incrementales, frecuentes y confiables, realizados por equipos DevOps multifuncionales que colaboran a lo largo del ciclo de vida del producto y se responsabilizan conjuntamente de los entregables.

El éxito de internet no habría sido posible sin el código abierto, que ha evolucionado desde proyectos comunitarios de base hasta convertirse en un movimiento generalizado. El próspero ecosistema de código abierto actual ha proporcionado una infraestructura segura y escalable, que constituye la columna vertebral de internet y de la mayoría de los entornos de aplicaciones empresariales modernos. La comunidad de código abierto ha proporcionado sistemas operativos (p. ej., Linux), infraestructura de contenedores (p. ej., Kubernetes y Docker), middleware para microservicios (p. ej., Swagger) y la cadena de herramientas para CI/CD. Entre las organizaciones de código abierto líderes en este ámbito se incluyen la Fundación Linux, la Fundación Eclipse y la Fundación Cloud Native Computing. La industria de TI aprendió que, al colaborar en componentes no diferenciadores, como el kernel de Linux (compartido por todo, desde televisores inteligentes hasta servidores en la nube), se pueden destinar más recursos a los componentes diferenciadores y orientados al cliente de sus productos. Hoy en día, el código abierto se ha convertido en un mercado multimillonario que incluye

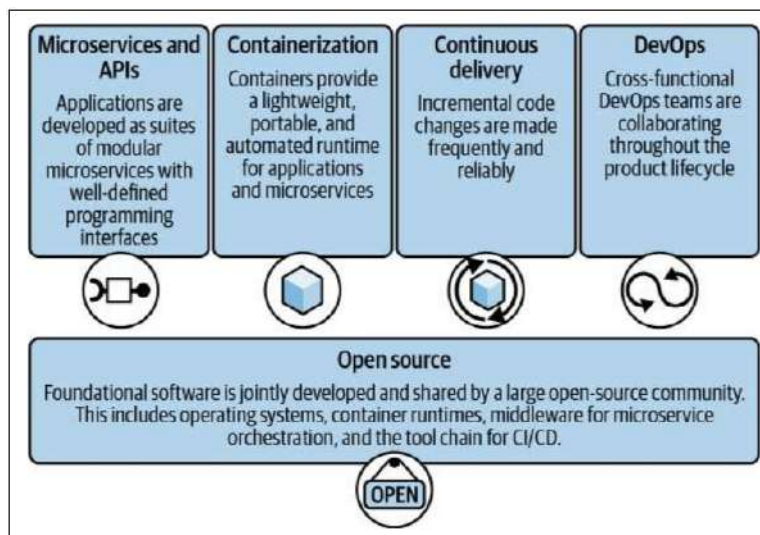


Figura 2.2: Elementos clave del desarrollo nativo de la nube y del código abierto

soporte empresarial comercial, consultoría, personalización, capacitación, servicios de alojamiento, licencias duales y creación de productos comerciales sobre bases de código abierto.

3

Sistema operativo (SO) vehicular y tecnologías facilitadoras

Casi todos los fabricantes de equipos originales (OEM), y muchos de sus proveedores, llevan tiempo trabajando en la creación de un sistema operativo (SO) vehicular moderno. Aún no existe una definición completa y ampliamente aceptada de lo que comprende un SO vehicular. Sin embargo, la mayoría de las definiciones incluyen los siguientes componentes clave: entornos de ejecución de software especializados para diferentes dominios funcionales, desacoplamiento de hardware y software mediante API estandarizadas y actualizaciones de software inalámbricas (OTA).

Según esta definición, un SDV es un componente clave del sistema operativo del vehículo, o incluso un sinónimo. Los SDV se habilitan mediante un conjunto de tecnologías eficientes y continuas, incluso después del inicio de la producción (SOP) y la entrega de nuevas funciones digitales del vehículo, que combinan componentes integrados y externos en una arquitectura de software integral. Comenzaremos analizando la arquitectura eléctrica y electrónica (E/E) emergente y su compatibilidad con una arquitectura orientada a servicios (SOA). Los elementos clave de la E/E y la SOA son la abstracción de hardware, las API del vehículo y la pila tecnológica del SDV. Los vehículos modernos utilizan actualizaciones OTA para soportar las actualizaciones posteriores al SOP, que eventualmente sentarán las bases de las tiendas de aplicaciones para vehículos. Finalmente, analizaremos cómo la inteligencia artificial (IA) puede mejorar el vehículo centrado en software.

Fundamento: Arquitectura E/E

Hoy en día, la denominada arquitectura E/E describe el diseño y la disposición general de los sistemas eléctricos y electrónicos de un vehículo. Esta arquitectura abarca la distribución de energía, datos y señales de control en todo el vehículo, así como la integración e interconexión de diversos componentes y sistemas E/E. Muchos vehículos aún implementan una arquitectura E/E centrada en el dominio, donde los diferentes dominios del vehículo, como el tren motriz, el chasis, el habitáculo y la carrocería, se

agrupan lógicamente y se conectan mediante sistemas de bus dedicados, como el bus CAN (Red de Área del Controlador). El bus CAN es un protocolo basado en señales diseñado para permitir que las unidades de control electrónico (ECU) y otros nodos de cómputo de un vehículo se comuniquen entre sí de forma fiable y prioritaria.

Dado que la arquitectura de dominio E/E genera arneses de cableado vehiculares muy complejos y pesados, los fabricantes de equipos originales (OEM) utilizan las denominadas arquitecturas zonales, cuyo objetivo es agrupar los diferentes sensores y actuadores del vehículo según su ubicación física en el mismo. En una arquitectura E/E zonal, los arneses de cableado se vuelven menos redundantes, lo que permite conexiones simplificadas dentro de cada zona, reduce la complejidad y el peso, y facilita la integración de nuevas funciones y tecnologías. Las arquitecturas zonales suelen combinar controladores de zona dedicados con ordenadores de vehículo de alto rendimiento. Los controladores de zona se conectan localmente a diversos sensores y actuadores, a menudo utilizando diferentes sistemas de bus heredados, como CAN, Red de Interconexión Local (LIN) y FlexRay. Los controladores de zona se conectan entre sí y con los ordenadores de vehículo de alto rendimiento a través de nuevas redes integradas de alta velocidad basadas en Ethernet (la base del Internet actual).

API de vehículos

El primer paso hacia una arquitectura orientada a servicios para servicios digitales a bordo y fuera de él es proporcionar una abstracción de hardware mediante las API de vehículos. Hoy en día, el desarrollo de nuevas funciones a bordo suele implicar un proceso de alineación complejo y prolongado entre varios departamentos del fabricante de equipos originales (OEM). Esto se debe a que todas las señales dentro de un dominio a bordo determinado se comunican a través de un sistema de bus compartido (por ejemplo, el bus CAN). Para cada tipo de vehículo, una matriz CAN define qué ECU envía qué mensaje, bajo qué condiciones y con qué tiempo de ciclo, qué ECU recibe qué mensajes y cómo se estructuran y priorizan estos. Esto da como resultado una arquitectura estrechamente acoplada que requiere una alineación muy estrecha a nivel técnico y organizativo. Para pasar de aquí a una arquitectura orientada a servicios con un acoplamiento flexible, se requiere un nuevo nivel de abstracción de hardware.

Desde la perspectiva del consumidor de servicios (es decir, el usuario de aplicaciones digitales), el vehículo debe proporcionar un conjunto de API bien definidas que abstraigan sus funciones. En el nivel más bajo, estas son los sensores y actuadores del vehículo. Un buen ejemplo de este estándar abierto de la industria es la Especificación de Señales del Vehículo (VSS), definida por la **Alianza de Sistemas de Vehículos Conectados** (COVESA VSS). COVESA VSS define una estructura de API en forma de árbol para acceder a los sensores y actuadores del vehículo como señales (véase la Figura 3-1).

Por ejemplo, `Vehicle.Cabin.Seat.Row1.Pos1.Headrest.Angle` permitiría a un desarrollador de aplicaciones acceder al actuador que controla el ángulo del reposacabezas del asiento delantero izquierdo. Este nivel de abstracción es adecuado para desarrolladores acostumbrados al desarrollo nativo en la nube o para smartphones. Por supuesto,

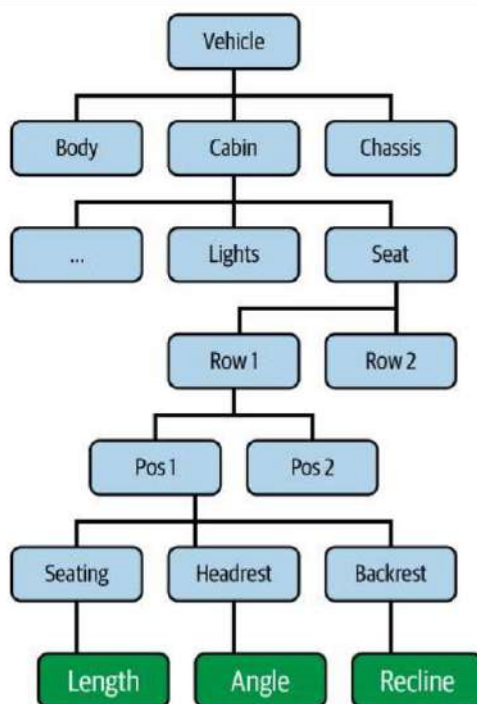


Figura 3.1: Árbol API de señales del vehículo (basado en COVESA VS)

estas API de señal a servicio de nivel relativamente bajo deben ampliarse con servicios de orquestación de nivel superior con el tiempo.

Sin embargo, este enfoque presenta algunos problemas. En primer lugar, como se mencionó anteriormente, puede resultar difícil mapear una API de software de tan alto nivel a la compleja arquitectura E/E subyacente. En segundo lugar, está la cuestión de cómo garantizar la seguridad funcional de las API que pueden afectar la física del vehículo. Y en tercer lugar, es necesario resolver los aspectos de calidad de servicio (QoS) de dicha API (por ejemplo, los requisitos en tiempo real). Analizaremos todos estos aspectos en las siguientes secciones.

SOA Vehicular: Encapsulando la Arquitectura E/E con APIs Vehiculares

El problema con la perspectiva de la arquitectura E/E es que se modela a partir del diseño físico del vehículo, incluyendo actuadores, sensores, redes a bordo y nodos de cómputo. Un requisito clave para los SDV será encapsular las arquitecturas E/E centradas en hardware con una arquitectura orientada a servicios (SOA) del vehículo, como se muestra en la Figura 3-2. En el lado del hardware E/E, las computadoras de alto rendimiento ejecutan las funciones SDV. Estas computadoras están conectadas físicamente con los demás componentes integrados, incluyendo nodos de cómputo más pequeños, sensores y actuadores. La traducción del hardware al software puede ocurrir en estas computadoras de alto rendimiento; las funciones de hardware se exponen mediante API. Estas API son la base de la comunicación entre los microservicios a nivel

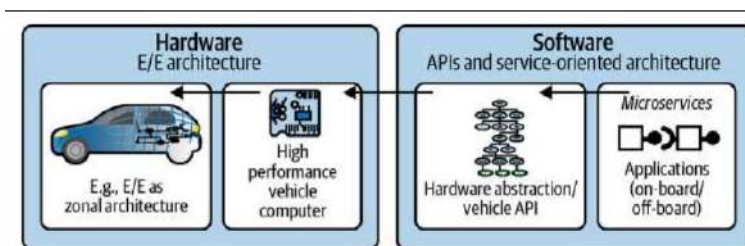


Figura 3.2: SOA que encapsula la arquitectura E/E

de aplicación y el hardware subyacente. Un microservicio es un software encapsulado que implementa una función específica. La interacción entre microservicios siempre ocurre mediante API. Estas pueden ser API que abstraen una función de hardware específica o API a nivel de aplicación. Una aplicación es un conjunto de microservicios orquestados para proporcionar una característica o servicio digital específico.

Capas de la SOA Vehicular

Las SOA introducen un nivel de capas arquitectónicas que ayuda a gestionar las características individuales de los diferentes microservicios involucrados. En el mundo de internet, por ejemplo, la capa frontend de una aplicación incluiría microservicios que cambian con frecuencia debido a la optimización continua de la interfaz de usuario, mientras que la capa de servicios básicos incluiría servicios más estables, centrados en datos, que cambian con mucha menos frecuencia. Este tipo de capas es esencial para la evolución eficiente de sistemas complejos.

En el mundo automotriz y SDV, esta estratificación es diferente y tiene múltiples dimensiones. La primera dimensión que debemos considerar es la integrada frente a la externa. La segunda dimensión se define por la seguridad funcional y los requisitos en tiempo real de los microservicios contenidos y las cadenas de eventos del vehículo encapsuladas por dichos microservicios.

Al combinar estas dimensiones con lo que comentamos anteriormente sobre las arquitecturas E/E y las API del vehículo, obtenemos una arquitectura orientada a servicios para el SDV, como se describe en la Figura 3-3. En la parte superior, la capa verde indica el entorno para los microservicios de solo gestión de calidad (QM) alojados en la nube. Un conjunto de API del vehículo a la nube conecta los servicios a bordo con los servicios externos. En la parte central, la capa SDV incluye microservicios de solo QM y ASIL A/B, alojados en diferentes entornos. Más al sur en esta arquitectura, se encuentran las funciones físicas del vehículo, accesibles mediante una API de señal a servicio (por ejemplo, basada en COVESA VSS). La API de señal a servicio debe encapsular la asignación de las API a la arquitectura E/E del vehículo. Por ejemplo, la API debe saber qué controlador de zona alojará las funciones necesarias para soportar `Vehicle.Cabin.Seat.Row1.Pos1.Headrest.Angle` (véase la Figura 3-1). El controlador de zona (en este caso la “Zona FL” de la parte delantera izquierda) necesitaría, a su vez, proporcionar un proxy de software que pueda traducir la API a la señal CAN correspondiente para cambiar el ángulo del reposacabezas delantero

izquierdo.

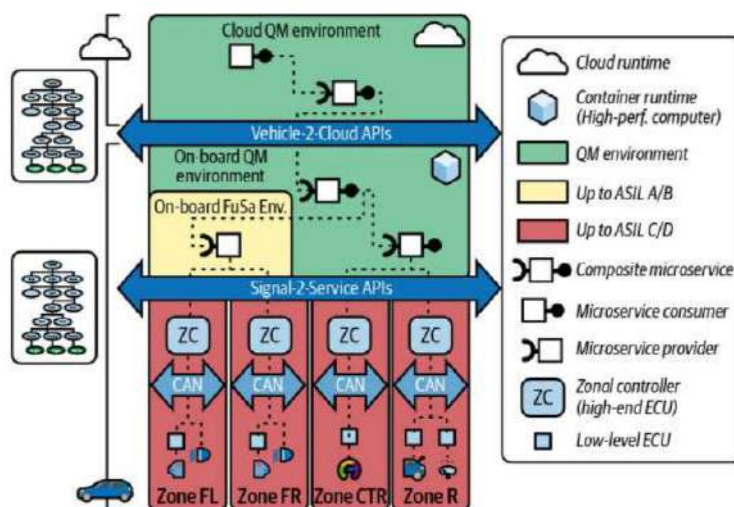


Figura 3.3: Arquitectura orientada a servicios para el vehículo definido por software

Garantizando la Seguridad Funcional en la SOA Vehicular

Una ventaja clave de la arquitectura de servicios vehiculares multicapa es que ahora disponemos de varias opciones para la seguridad funcional. Veamos un ejemplo concreto en la Figura 3-4: una función en una aplicación para smartphones que abre el maletero de un vehículo de forma remota.

El problema con la función de "abrir el maletero" es que solo puede ejecutarse de forma segura si el vehículo se encuentra en un estado seguro (es decir, sin moverse). Por lo tanto, la pregunta es: ¿quién realizará esta comprobación y cómo? El usuario no puede realizarla por sí mismo; esto violaría los principios de acoplamiento flexible de la arquitectura orientada a servicios, ya que la implementación del servicio no puede asumir que los clientes siempre siguen correctamente un protocolo determinado. Esto es cierto tanto para los clientes que llaman desde la nube como para los que llaman desde un entorno QM a bordo del vehículo. De hecho, este es el objetivo principal del entorno QM: no proporciona ninguna de las propiedades de QoS de ASIL. Esto significa que el servicio de "abrir el maletero" a bordo debe realizar las comprobaciones de seguridad necesarias por sí mismo. Suponiendo que la interfaz de servicio esté expuesta a los clientes de QM, ahora hay al menos dos opciones.

En la primera opción (Figura 3-4, izquierda), el servicio integrado de "apertura del maletero" (Vehicle.Body.Trunk.Rear.IsOpen) se ejecuta en un entorno QM y, por lo tanto, no se considera seguro. Podría proporcionar algunos servicios adicionales, pero la comprobación de seguridad debe realizarse desde el entorno de seguridad ASIL. En nuestro caso, este sería el controlador de zona del maletero. Aquí, el sistema debe comprobar la velocidad actual del vehículo antes de abrir el maletero. Sin embargo,

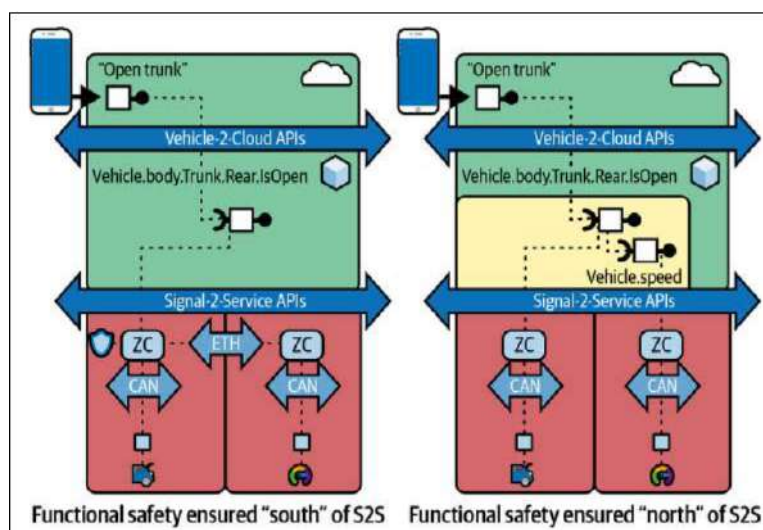


Figura 3.4: SOA vehicular y seguridad funcional: dos ejemplos

en este ejemplo, un controlador de zona diferente gestiona la señal de velocidad (es decir, el controlador de zona del tren motriz). Por lo tanto, el controlador de zona que gestiona el maletero debe acceder al controlador de zona que gestiona la señal de velocidad antes de comunicarse con la ECU de bajo nivel que controla el maletero. Todo esto debe ocurrir en tiempo real (por eso esta comprobación se realiza en el controlador de zona, no en la capa QM superior). Esto no significa necesariamente que deba ocurrir en una fracción de milisegundo, sino dentro de un intervalo de tiempo garantizado. Para que esto suceda, los dos controladores de zona deben estar en comunicación más directa, como un enlace directo que admita la denominada red sensible al tiempo (TSN) a través de Ethernet de alta velocidad.

La segunda opción (Figura 3-4, derecha) asume que el servicio de "troncal abierto" integrado se ejecuta en un entorno de seguridad que cumple con los niveles de ASIL requeridos para esta función. Esto significa que la interfaz expuesta aún puede ser invocada desde un cliente QM (por ejemplo, una secuencia de eventos originada en la nube), pero la implementación detrás de la interfaz es segura. Si existe otro servicio *Vehicle.Speed* que cumpla con las normas de seguridad (y una forma segura para que el servicio troncal invoque el servicio de velocidad), todo esto puede realizarse en el mismo entorno. La ventaja es que el servicio de velocidad se proporciona como un microservicio real que puede ser reutilizado tanto por QM como por los servicios ASIL. Sin embargo, esto requiere una arquitectura de servicio capaz de orquestar microservicios con niveles de QoS ASIL.

Ambas opciones presentadas aquí son válidas. La primera opción exige menos niveles de integración dentro del entorno y podría ser más fácil de implementar con las tecnologías actuales. La segunda opción tiene mayor potencial para habilitar casos de uso avanzados entre dominios, pero requiere un entorno de ejecución más avanzado y compatible con ASIL. A continuación, se analizará la pila tecnológica de SDV necesaria para ambas opciones.

Pila tecnológica SDV para la arquitectura SOA vehicular

Para comprender cómo una arquitectura SOA vehicular puede soportar capas arquitectónicas con diferentes niveles de QoS, necesitamos añadir otra dimensión: el hardware, el sistema operativo y el entorno de middleware/aplicación. Esto se muestra en la Figura 3-5.

La capa de la nube que se muestra aquí es la esperada. El hardware incluye unidades centrales de procesamiento (CPU) genéricas y unidades de procesamiento gráfico (GPU), que se utilizan actualmente para procesar cargas de trabajo de aprendizaje automático en la nube. A nivel de sistemas operativos (SO), solemos encontrar un SO de propósito general como Linux. Los hipervisores o máquinas virtuales se utilizan para ejecutar múltiples instancias del SO en hardware compartido, lo cual es importante para la escalabilidad y la elasticidad de la nube. Las nubes modernas también proporcionan middleware y entornos de ejecución de aplicaciones muy completos. A continuación,

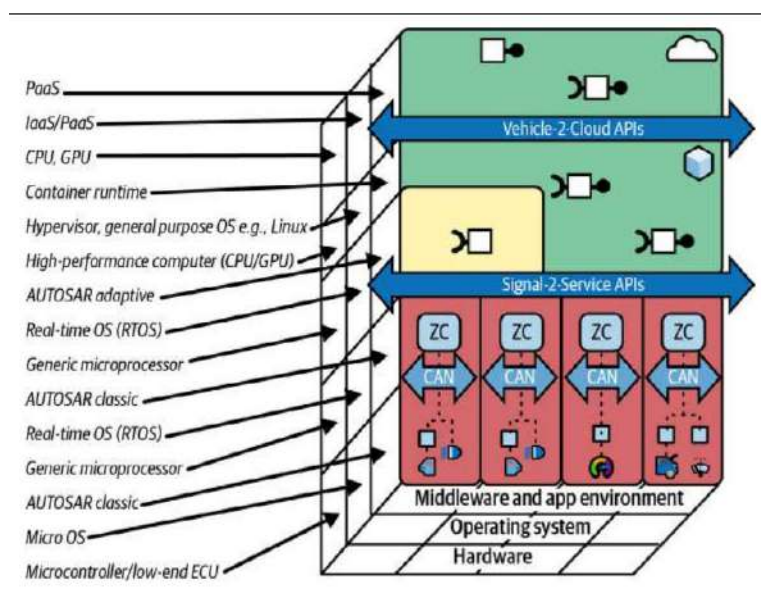


Figura 3.5: La pila tecnológica de SDV

se encuentra el entorno integrado para microservicios y aplicaciones de QM. Este tipo de entorno busca replicar al máximo la rica pila tecnológica disponible actualmente en la nube, añadiendo además aspectos específicos del vehículo. Estos incluyen, por ejemplo, tiempos de arranque más rápidos, eficiencia energética y compatibilidad con la gestión de instancias altamente distribuidas que no siempre están en línea (por ejemplo, grandes flotas de vehículos, en lugar de centros de datos en la nube). El sistema operativo de esta capa suele ser un sistema operativo de propósito general combinado con hipervisores para la virtualización. Este se ejecutará en un ordenador de vehículo genérico de alto rendimiento, posiblemente con GPU adicionales para cargas de trabajo de IA/ML.

La siguiente capa debe soportar altos niveles de QoS y, por consiguiente, requiere un entorno de alta disponibilidad más tradicional, generalmente con soporte en tiempo

real. Un sistema operativo ampliamente utilizado en esta área es QNX de BlackBerry. Los controladores de zona probablemente compartirán una configuración similar. Se requerirá un middleware con capacidad en tiempo real para vincular los microservicios del entorno al norte de la API S2S con aquellos que residen al sur de esta (por ejemplo, en los controladores de zona).

Finalmente, los controladores de zona se conectan a las ECU de nivel inferior. Especialmente para el control de sensores y actuadores específicos del vehículo, se utilizan microcontroladores. Estos son circuitos integrados (CI) que suelen representar un sistema en chip (SoC) completo, que incluye el núcleo del procesador, la memoria y las E/S, todo dentro de un encapsulado discreto. Aquí se utilizan entornos especializados para funciones integradas de bajo nivel, pero altamente eficientes y específicas. Entre el controlador de zona y las diferentes ECU y microcontroladores de bajo nivel, se deben soportar diferentes sistemas de bus, como CAN, LIN y FlexRay.

OTA: Actualizaciones inalámbricas

Anteriormente, la industria automotriz buscaba congelar el hardware y el software de soporte de una nueva generación de vehículos en el SOP. Los cambios posteriores al SOP en el software de soporte solían realizarse solo si era necesario solucionar problemas graves.

De las 70 a 100 ECU y controladores que suelen encontrarse en un vehículo tradicional, la mayoría nunca se actualizaba tras la actualización inicial del software como parte del proceso de fabricación. Si se necesitaban actualizaciones, se ejecutaba una campaña de actualización específica para cada variante afectada ("push"). Las actualizaciones suelen empaquetarse manualmente tras una exhaustiva investigación y validación de variantes.

En el futuro, muchos fabricantes de equipos originales (OEM) visualizan un rico ecosistema de aplicaciones en desarrollo para sus vehículos. Las actualizaciones de software ya no se realizarán solo por razones de calidad, sino que podrán realizarse bajo demanda. Los clientes seleccionarán qué nuevas aplicaciones y funciones desean descargar y activar ("pull"). Estas nuevas aplicaciones pueden abarcar diferentes ámbitos, desde el infoentretenimiento y el bienestar hasta la comodidad del habitáculo y el rendimiento de la conducción. Sin embargo, esta creciente individualización también conllevará un aumento drástico de las variantes de combinaciones de hardware y software. La próxima generación de OTA deberá abordar este problema (Figura 3-6).

La tienda de aplicaciones para vehículos

El santo grial del SDV es la tienda de aplicaciones para vehículos. La industria de los smartphones ha demostrado el enorme potencial de las aplicaciones y el contenido generado por socios y desarrolladores externos, accesibles a los clientes bajo demanda y después del procedimiento operativo estándar (SOP). Ahora, los fabricantes de equipos originales (OEM) se están sumando a la tendencia. Para replicar el éxito de las tiendas de aplicaciones para smartphones en la industria automotriz, se requieren

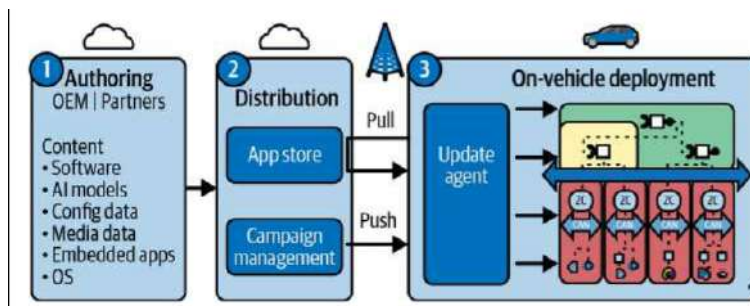


Figura 3.6: Arquitectura moderna de actualización inalámbrica

varios aspectos. La Figura 3-7 ofrece una descripción general de los requisitos técnicos. En primer lugar, los vehículos necesitan un entorno de ejecución de aplicaciones a

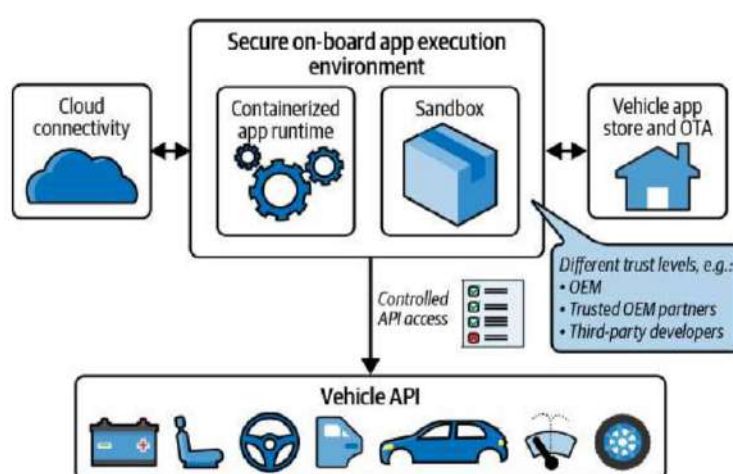


Figura 3.7: Tienda de aplicaciones para vehículos

bordo seguro para la ejecución de las aplicaciones descargadas a través de la tienda de aplicaciones del vehículo. Este entorno de ejecución se conoce a menudo como "sandbox". Es importante que este "sandbox" sea seguro, impidiendo que el código se filtre y acceda a otras partes del vehículo. Un virus que tome el control de un vehículo puede tener consecuencias mortales para los ocupantes o transeúntes.

En segundo lugar, la arquitectura del sistema debe garantizar que las aplicaciones que se ejecutan dentro del "sandbox" solo puedan interactuar con otras partes del vehículo a través de un conjunto de API controladas. Lo más probable es que los fabricantes de equipos originales (OEM) tengan que definir diferentes niveles de confianza para sus API, diferenciando entre el acceso a las API a través de aplicaciones desarrolladas por ellos mismos, aplicaciones de socios de confianza y, potencialmente, aplicaciones desarrolladas por terceros completamente desconocidos. Algunos vehículos disponibles actualmente ya ofrecen tiendas de aplicaciones para el infoentretenimiento a bordo. En el futuro, las tiendas de aplicaciones para vehículos también deberían proporcionar acceso a las API de sensores para impulsar la creatividad de la comunidad global de desarrolladores. Incluso podrían ofrecer acceso seguro, por lo que ciertos actuadores podrían ser una opción (por ejemplo, en el ámbito de la carrocería y la comodidad).

Si bien los gigantes de los smartphones ya podían posicionar aplicaciones a través de Apple CarPlay y Android Automotive, muchos fabricantes de equipos originales (OEM) están empezando a ofrecer tiendas de aplicaciones para aplicaciones que se ejecutan de forma nativa en el vehículo. Este proceso comienza en el área del infoentretenimiento, con la introducción de aplicaciones diseñadas para ejecutarse de forma nativa en la pantalla del sistema de infoentretenimiento integrado. Algunos ejemplos incluyen aplicaciones de música y podcasts, videoconferencias, información meteorológica, juegos, noticias, aparcamiento y carga de vehículos eléctricos (VE). Queda por ver qué tan bien estas aplicaciones de infoentretenimiento para vehículos podrán competir con sus homólogas para smartphones.

Para diferenciarse mejor, es probable que la próxima generación de aplicaciones para vehículos utilice funciones específicas del vehículo. Una vez que las aplicaciones puedan interactuar de forma significativa y segura con el vehículo, se podrá crear una nueva experiencia. Acceder a los datos de los sensores del vehículo, e incluso controlar algunos de sus actuadores, creará un nuevo tipo de aplicación. Por ejemplo, acceder a sensores del vehículo, como cámaras, radares y micrófonos, podría dar lugar a una nueva generación de aplicaciones de salud y bienestar. Esto ayudaría a los fabricantes de equipos originales (OEM) a acercarse a la visión de un "hábitat sobre ruedas".

El SDV y la IA

Si bien la IA ya se ha convertido en un tema candente, el lanzamiento de ChatGTP ha trasladado el debate de "el software se comerá el mundo" a "la IA se comerá el mundo". La IA es disruptiva en muchos sentidos, y el sector automotriz no es la excepción. Entonces, ¿necesitamos un "vehículo definido por la IA" en lugar de (o además de) uno definido por software? ¿En qué parte de la arquitectura SDV juega un papel la IA? La respuesta es: potencialmente en todas las capas de la arquitectura SDV (véase la Figura 3-8). En el smartphone, la IA puede utilizarse para controlar el

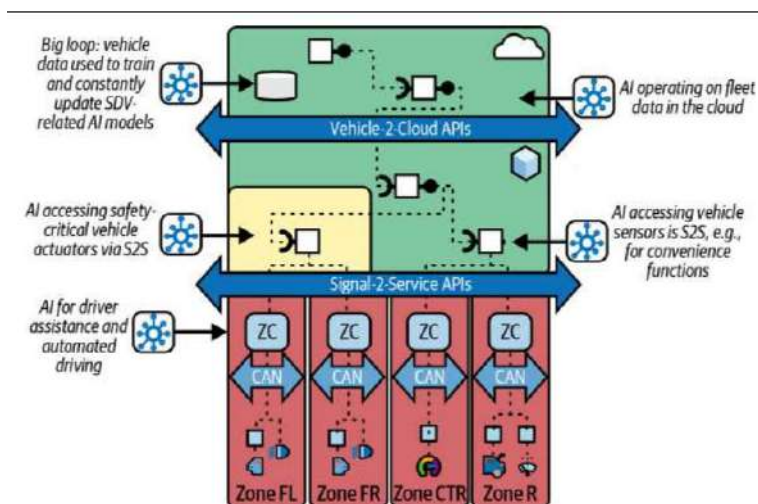


Figura 3.8: El SDV y la IA

acceso al vehículo o funciones específicas (p. ej., mediante reconocimiento facial). En

la nube, la IA puede operar con datos de la flota de vehículos (p. ej., para facilitar la planificación de rutas, optimizar la carga de vehículos eléctricos, realizar análisis de tráfico, mejorar el rendimiento de la batería, etc.).

A bordo, observamos diferentes tipos de IA. Para la conducción altamente automatizada, la IA está profundamente integrada con la arquitectura E/E del vehículo, proporcionando identificación de objetos (p. ej., reconocer a un niño en bicicleta delante del coche) y controlando la trayectoria del vehículo (p. ej., frenar al niño mencionado). Además, parece existir un enorme potencial para las aplicaciones de larga duración basadas en IA. Estas aplicaciones requieren una inversión mucho menor para su desarrollo y probablemente tendrán un impacto general menor individualmente, pero aun así tendrán un gran impacto global. Las tiendas de aplicaciones para smartphones han demostrado que, si las barreras de entrada son lo suficientemente bajas, se creará una gran cantidad de aplicaciones nuevas y creativas. Esto también debería ser válido para las aplicaciones de vehículos habilitadas con IA.

La combinación de la IA con los datos de los sensores del vehículo podría ser revolucionaria (por ejemplo, en el ámbito del infoentretenimiento y el bienestar). Si esto se combinara con funciones compatibles con SDV, se podría crear una nueva generación de aplicaciones para vehículos (por ejemplo, aplicaciones de confort en el habitáculo que utilizan datos de sensores para modificar el ambiente del vehículo accediendo a la iluminación, la climatización (HVAC, en inglés) y las funciones de masaje de los asientos, entre otras). El uso de la IA llegará mucho más allá de las aplicaciones para vehículos. Desde la asistencia por voz hasta el mantenimiento predictivo, desde las operaciones de flotas hasta la conducción automatizada y las cadenas de herramientas de desarrollo asistidas por IA, la IA revolucionará la forma en que diseñamos, desarrollamos, operamos e interactuamos con los vehículos.

4

Gestión del Flujo de Valor para el SDV

Internet se basa en tecnologías sofisticadas y en constante evolución. Por consiguiente, muchos negocios exitosos en internet se basan en una cultura tecnológica. Sin embargo, esta dependencia de la tecnología a veces puede dificultar centrarse en el desarrollo de productos centrados en el cliente, el aumento de la competitividad y la generación de ingresos. Por ello, muchas empresas han adoptado la Gestión del Flujo de Valor (VSM) - Value Stream Mangement - como una buena práctica para sus negocios digitales.

La VSM es un conjunto de prácticas diseñadas para garantizar que las nuevas funcionalidades digitales se implementen de forma rápida y eficiente, y que aporten un valor claro al cliente.

Entonces, ¿cómo se puede aplicar la VSM al vehículo definido por software? Veamos un poco.

Trabajando a Diferentes Velocidades

Quizás la conclusión más importante en el contexto del SDV es que no puede haber un único flujo de valor. Nuestro análisis del "choque de dos mundos" y las posteriores discusiones técnicas han demostrado que diferentes requisitos requieren enfoques especializados. La experiencia digital del vehículo debe proporcionarse mediante un flujo de valor digital que se adhiera a las mejores prácticas ágiles, mientras que la experiencia física del vehículo debe proporcionarse mediante un flujo de valor que se adhiera al enfoque riguroso y de "acierto a la primera" requerido para áreas con altos niveles de seguridad funcional (véase la Figura 4-1). El flujo de valor digital debe ser capaz de abordar territorios inexplorados, gestionar requisitos imprecisos e ideas en constante cambio, y reaccionar rápidamente a la retroalimentación de los clientes. Desarrollar un enfoque exploratorio y basado en la retroalimentación para el flujo de valor digital es clave. Los canales de entrega técnica del flujo de valor digital generarán artefactos que se implementarán en la nube o a bordo. Los artefactos a bordo utilizarán SDV, contenedorización, OTA y API de vehículos. Los mecanismos para medir el éxito del cliente pueden integrarse de forma nativa en esta pila tecnológica, de forma similar a las herramientas de análisis del recorrido del cliente habituales en las aplicaciones de

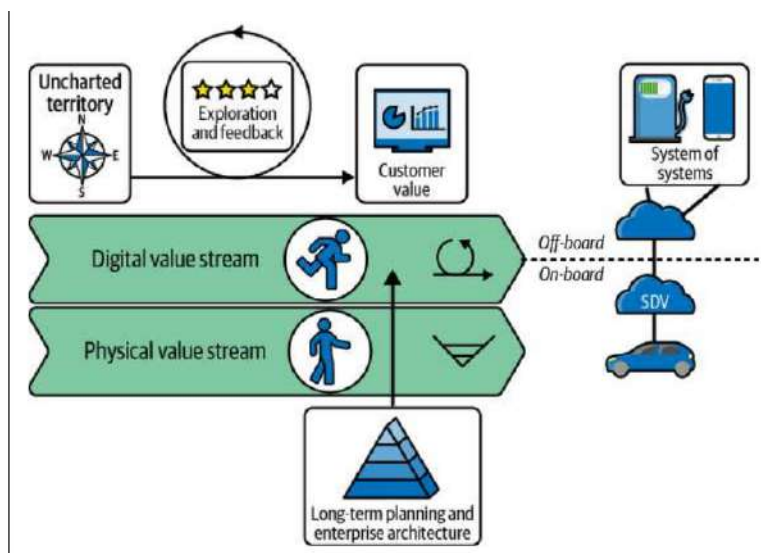


Figura 4.1: Perspectiva de flujo de valor de alto nivel para el sdv

internet.

El flujo de valor físico debe estar mucho más alineado con la planificación a largo plazo y la arquitectura empresarial. Los métodos aplicados se ceñirán a las mejores prácticas establecidas para las características funcionales relacionadas con la seguridad (por ejemplo, el consolidado modelo V de desarrollo de software, que incorpora mecanismos formales de verificación y validación). Los resultados técnicos del flujo de valor físico incluirán una combinación de hardware y software, como software integrado y unidades de control electrónico (ECU), así como componentes del sistema mecatrónico. Implementar mecanismos para medir el éxito del cliente en el flujo de valor físico será más difícil.

Establecer una estrategia eficaz de VSM que permita a los fabricantes de equipos originales (OEM) trabajar a diferentes velocidades en distintas áreas será un factor clave para el éxito en el futuro.

Divide y vencerás

La ventaja de SOA y la metodología de trabajo centrada en API que presentamos anteriormente reside en que estos enfoques nos permiten crear integración no solo a nivel técnico, sino también a nivel organizativo. Para respaldar flujos de valor que se mueven a diferentes velocidades, se requiere un acoplamiento flexible tanto a nivel técnico como organizativo. Y esto es precisamente lo que se puede lograr con las API y la abstracción de hardware.

La Figura 4-2 muestra un ejemplo de acoplamiento flexible. En la parte superior, se crean artefactos técnicos como resultado del flujo de valor digital. Primero, se crean e implementan prototipos con la API del vehículo; por ejemplo, utilizando el [digital.auto.playground](#) para la creación rápida de prototipos en línea de las características del SDV. Estos prototipos pueden utilizarse para obtener retroalimentación temprana

de las partes interesadas clave, incluyendo clientes y la gerencia. Posteriormente, los primeros prototipos de SDV se perfeccionan y se acercan a las aplicaciones reales. En las primeras fases de desarrollo, estas aplicaciones SDV pueden utilizar simulaciones de vehículos subyacentes a las API para crear entornos de prueba realistas. Una vez disponible el hardware del vehículo real, este puede reemplazar la simulación del vehículo. Dado que las API no cambian (al menos en un escenario ideal) y la aplicación SDV se implementa sobre la API, esta transición de una simulación de vehículo a un hardware de vehículo real debería ser fluida desde el punto de vista de la aplicación SDV. Esto significa que la API se convierte en el mecanismo que garantiza una conexión flexible no solo entre los artefactos técnicos, sino también entre las diferentes organizaciones involucradas. Los equipos que desarrollan el flujo de valor digital lo hacen sobre las API que representan el hardware real subyacente (de ahí la “abstracción de hardware”), y mediante el uso de simuladores, pueden avanzar a su propio ritmo, independientemente de la disponibilidad del hardware real. Este enfoque lleva los conceptos consolidados de hardware en el ciclo (HIL) y software en el ciclo (SIL) a otro nivel. Poder dividir un conjunto complejo de tareas mediante una estrategia de

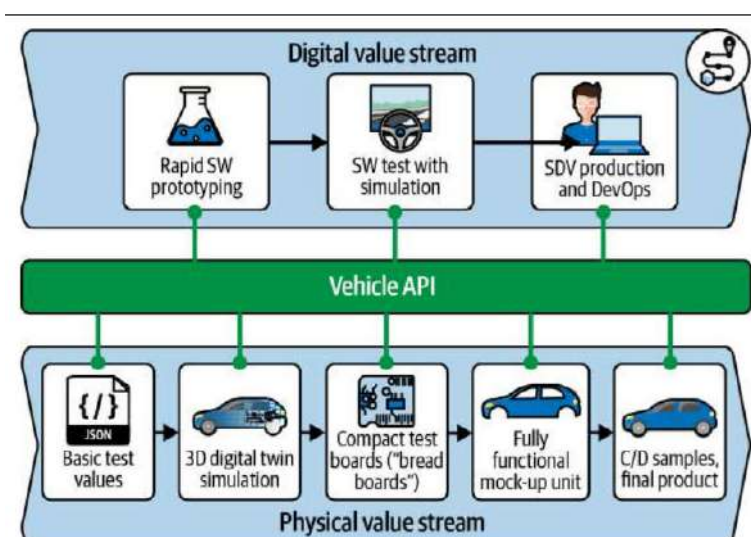


Figura 4.2: Flujos de valor con artefactos técnicos que se mueven a diferentes velocidades

“divide y vencerás” supone una gran ventaja, ya que reduce la complejidad a un nivel manejable. Cabe destacar que las API probablemente actúen como un “reloj maestro”, ayudando a sincronizar el trabajo entre los diferentes equipos en los distintos flujos de valor.

Perspectiva Empresarial

Tradicionalmente, los fabricantes de equipos originales (OEM) han abordado la complejidad a la que se enfrentan mediante la gestión de la arquitectura empresarial (EAM) y la ingeniería de sistemas basada en modelos (MBSE), como se muestra en la Figura 4-3. EAM ayuda a gestionar las dependencias entre la perspectiva de sistemas de sistemas (p. ej., el vehículo en el contexto de su entorno), la perspectiva del sistema (el vehículo en sí) y los subsistemas, incluyendo componentes y características clave. Por

supuesto, todo esto debe considerarse en el contexto de las diferentes variantes y tipos de vehículos. Finalmente, gestionar la reutilización de las plataformas de vehículos es fundamental, incluyendo las plataformas de hardware, las plataformas E/E y las plataformas de software. MBSE desempeña un papel cada vez más importante en el diseño detallado de muchos componentes del sistema y sus interdependencias.

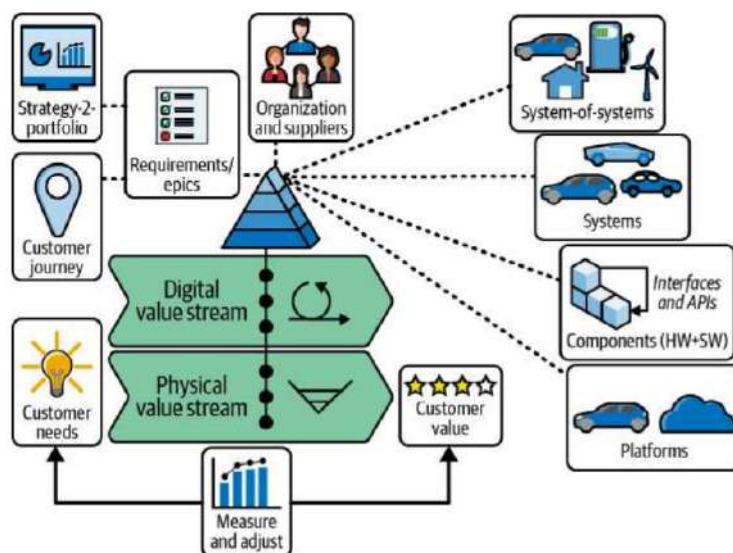


Figura 4.3: La perspectiva empresarial

Sin embargo, volviendo al debate sobre la "confusión de dos mundos", es importante destacar que este tipo de herramientas y métodos suelen considerarse muy controvertidos en el mundo del software, que ha pasado los últimos 20 años adoptando una cultura ágil. Esto se aprecia claramente en la Tabla 4-1, que compara cómo los valores ágiles definidos en el famoso **manifiesto para el desarrollo de software Ágil** se corresponden con los enfoques centrados en modelos y en código para el desarrollo automotriz.

Como ya hemos comentado, la capacidad de trabajar con diferentes flujos de valor que incorporan distintos enfoques y métodos es clave para el éxito. A nivel empresarial, se deben establecer métodos y mecanismos para mantener sincronizados los diferentes flujos de valor, con el apoyo de un enfoque flexible a nivel organizativo. De nuevo, las API pueden desempeñar un papel clave en este contexto, por ejemplo, al proporcionar una conexión flexible entre la perspectiva ágil/centrada en el código y la perspectiva centrada en el modelo/MBSE.

Valores del Manifiesto Ágil	Centrado en el modelo (EAM/MBSE)	Centrado en el código/SDV
Individuos e interacciones sobre procesos y herramientas	Herramientas y procesos necesarios, especialmente para hardware y software con requisitos ASIL	Ideal para procesos ágiles de gestión de calidad.
Software funcional con documentación exhaustiva	Se puede lograr mediante modelos ROM generados por código (pero no es una tarea trivial)	Bien apoyado
Colaboración con el cliente por encima de la negociación del contrato	Modelo como contrato; la validación temprana del cliente requiere exploración virtual	API como contratos, pero funciones desarrolladas en estrecha colaboración con los clientes
Responder al cambio en lugar de seguir un plan	Se requiere planificación a largo plazo para el hardware y el software ASIL	Con el apoyo del enfoque ágil

Cuadro 4.1: Desarrollo centrado en el modelo versus desarrollo centrado en el código

5

#digitalfirst: Una Nueva Forma de Trabajar

Considerando todo lo que hemos discutido hasta ahora, proponemos una nueva forma de trabajar para los OEM, a la que simplemente llamamos #digitalfirst - Lo digital primero-.

La naturaleza cambiante de las preferencias de los consumidores actuales implica que no podemos predecir con certeza qué características serán demandadas mañana. Sin embargo, surge una certeza clave: los OEM que no puedan explorar, probar y ofrecer nuevas características a gran escala con rapidez podrían verse incapaces de crear las experiencias atractivas que los consumidores actuales demandan. En nuestra era digital en rápida evolución, la agilidad para innovar con rapidez y eficacia no solo es deseable, sino esencial para mantener su relevancia en la industria automotriz.

En consecuencia, #digitalfirst comienza con las experiencias del cliente y se proyecta hacia atrás, hasta llegar a la tecnología. Para apoyar esto, cofundamos la iniciativa `digital.auto`. "Digital first, auto second" no es solo un eslogan, sino el lema fundamental que guía esta transformación. Como se muestra en la Figura 5-1, #digitalfirst asume que un OEM debe experimentar tres cambios radicales: el cambio hacia el norte (de la API del vehículo), el cambio hacia la izquierda (hacia las pruebas iniciales) y el cambio hacia el desarrollo virtualizado. Analizaremos cada uno de estos cambios con más detalle.

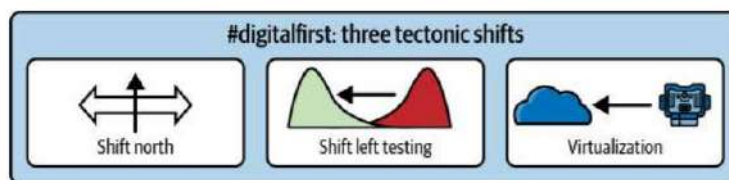


Figura 5.1: Los tres cambios tectónicos que subyacen a #digitalfirst

Desplazamiento hacia el Norte

Este desplazamiento hacia el norte implica trasladar funcionalidades del ámbito de la seguridad al de la gestión de calidad, creando una separación entre ambos mediante API bien definidas. Esto se debe al esfuerzo adicional que implica cada desarrollo en el ámbito de la seguridad (validación, homologación y documentación adicional), que normalmente se requiere en menor medida en el ámbito de la gestión de calidad.

Trasladar el código hacia el ámbito de la gestión de calidad, como se muestra en la Figura 5-2, permite utilizar técnicas y herramientas modernas de ingeniería de software, lo que acelera el desarrollo y facilita considerablemente las actualizaciones posteriores a los procedimientos operativos estándar (SOP). Además, en este ámbito, incluso los desarrolladores de software sin años de experiencia en el sector automotriz (los ingenieros de software modernos que también conocen la norma ISO 26262 son escasos) pueden trabajar de forma productiva, acelerando así significativamente el desarrollo.

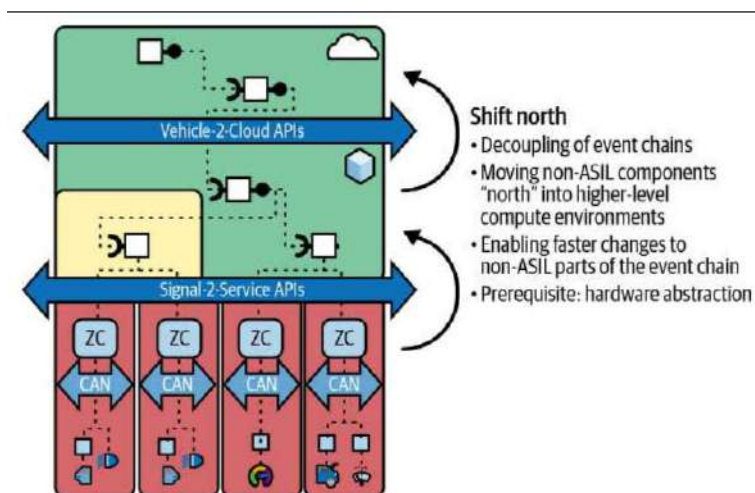


Figura 5.2: Desplazamiento hacia el Norte

Desplazamiento a la izquierda

El desplazamiento a la izquierda se refiere a explorar las características del cliente y realizar pruebas de usuario lo más pronto posible en el proceso de desarrollo, como se muestra en la Figura 5-3. Muchos de los coches actuales ofrecen una gran cantidad de botones y funciones, muchas de las cuales permanecen desconocidas o sin uso por los usuarios. Por lo tanto, el objetivo es identificar las características realmente deseables y maximizar la satisfacción del cliente mediante la exploración y las pruebas tempranas.

Según un informe del Instituto Nacional de Estándares y Tecnología (NIST), una agencia del Departamento de Comercio de EE. UU., el costo de solucionar problemas posteriores en el proceso de desarrollo puede ser hasta 640 veces mayor que el costo original de desarrollo. Y esto no incluye el costo de perder negocios debido a clientes insatisfechos o clientes que simplemente no se preocupan por ciertas características

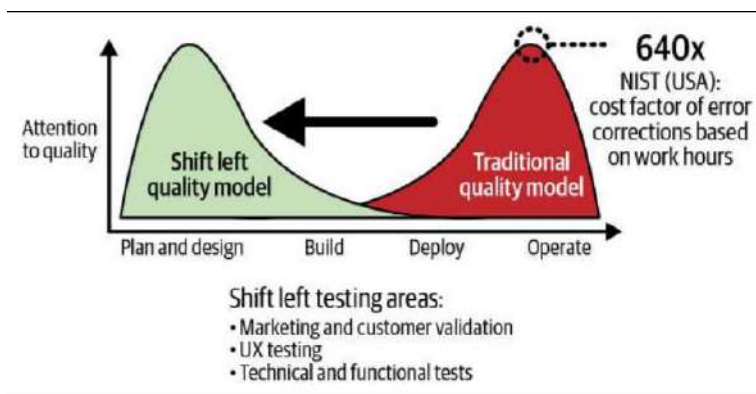


Figura 5.3: Prueba de desplazamiento a la izquierda

que los desarrolladores estaban seguros de que les encantarían, pero que nunca se molestaron en preguntar.

Virtualización

La virtualización se centra en el desarrollo y las pruebas de sistemas en entornos virtuales de nube. Una organización interesante en este ámbito es **SOAFEE**, que desarrolla conceptos de virtualización en el contexto de arquitecturas de hardware centradas en ARM. La principal motivación es superar la estrecha relación tradicional entre hardware y software en el desarrollo automotriz, donde los ingenieros de software deben esperar a versiones preliminares y costosas de hardware para su desarrollo y prueba. La integración y las pruebas de componentes son costosas y complejas debido a la escasez de prototipos y a las numerosas variaciones de los vehículos, como diferentes motores, niveles de equipamiento o requisitos específicos de cada país.

La capacidad que ofrecen los entornos de nube para una escalabilidad infinita y una reducción de costes, junto con el uso de unidades de control electrónico virtuales (vECU) o vehículos virtualizados, ofrece una solución a estos desafíos, como se muestra en la Figura 5-4.

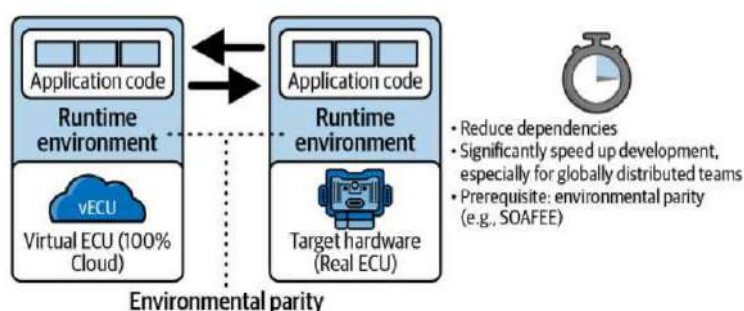


Figura 5.4: Virtualización

Resumen

Para hacer realidad la visión de un "hábitat sobre ruedas", con aplicaciones multidominio de alto rendimiento y una fusión de datos 10 veces más rápida, los fabricantes de equipos originales (OEM) deben superar los importantes impedimentos que presentan los requisitos de seguridad funcional, las limitaciones técnicas y las restricciones organizativas ("choque de dos mundos").

El sistema operativo del vehículo puede ser una plataforma potente que permite el desarrollo rápido de aplicaciones en el ámbito de la gestión de calidad (QM) mediante arquitecturas orientadas a servicios, OTA y tiendas de aplicaciones para vehículos. La combinación de SDV e IA es fundamental para las aplicaciones basadas en datos.

Para gestionar el desarrollo a diferentes velocidades, los OEM deben adoptar VSM y utilizar la abstracción de hardware y las API de vehículos para crear un acoplamiento flexible, no solo a nivel técnico, sino también a nivel organizativo, entre los flujos de valor digitales y físicos, como se muestra en la Figura 5-5. #digitalfirst es una nueva

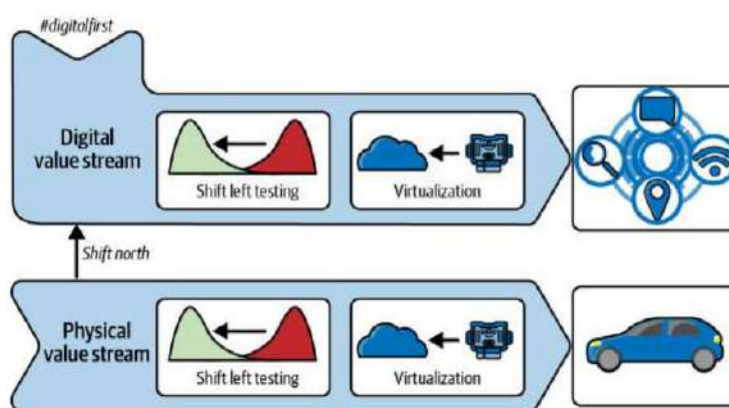


Figura 5.5: #digitalfirst y VSM para el digital.auto

forma de trabajar que combina tres cambios radicales:

Desplazamiento hacia el norte

Interrumpir las cadenas de eventos y centrarse en el desarrollo de aplicaciones QM al norte de la API del vehículo

Desplazamiento hacia la izquierda

Desplazamiento hacia la izquierda de las pruebas de usuario y obtención de una validación temprana del cliente de la experiencia digital/física integrada

Desplazamiento hacia la virtualización

Mayor desacoplamiento del desarrollo de hardware y software, aumentando así la agilidad

Juntos, SDV y #digitalfirst pueden ofrecer numerosos beneficios para los fabricantes de equipos originales (OEM) y toda la cadena de suministro. La retroalimentación

temprana y continua de los usuarios ayuda a garantizar que las inversiones digitales rindan los frutos previstos. Evitar inversiones en funciones digitales no deseadas ayuda a garantizar que las capacidades de desarrollo se utilicen donde generen el mayor valor para el cliente. Aumentar significativamente la velocidad de desarrollo (10 veces) proporciona la agilidad necesaria para reaccionar rápidamente a las cambiantes necesidades y preferencias de los clientes. El enfoque de "divide y vencerás", posible gracias a la abstracción del hardware y a los flujos de valor especializados, ayuda a gestionar la complejidad del mundo automotriz actual. La virtualización ayuda a reducir el coste y la complejidad de gestionar demasiados prototipos de hardware.

6

Siguientes pasos

Esperamos que los conceptos aquí descritos le resulten útiles en su trabajo diario. Si le interesa saber más sobre cómo trabajamos y cómo puede participar, visite [digital.auto](#).

Contribuimos a la creación conjunta de digital.auto como una comunidad abierta e independiente de proveedores para que nuestra industria pueda utilizar el SDV y ofrecer todos los interesantes casos de uso que hemos mencionado. Lo logramos mediante diversas actividades de colaboración y liderazgo intelectual, incluida esta publicación. Además, la comunidad de digital.auto ha colaborado para crear diversas actividades de código abierto.

La Figura 6-1 muestra las áreas de enfoque de digital.auto en relación con el ciclo de vida del SDV #digitalfirst, presentado en el Capítulo 1.

En primer lugar, nos centramos en el desarrollo de métodos y herramientas que respalden la exploración virtual inicial de las experiencias del SDV. En particular, colaboramos para crear un entorno de prototipado rápido basado en la nube para el SDV. Este entorno de pruebas digital.auto permite probar rápidamente nuevas ideas para el SDV con las API de vehículos reales, que se simulan en el backend para obtener datos de prueba realistas. Los prototipos resultantes permiten obtener retroalimentación temprana de los clientes y conocer mejor los requisitos, incluyendo las API necesarias para la nueva aplicación. Los prototipos desarrollados en el entorno de pruebas se pueden implementar directamente en plataformas SDV reales (por ejemplo, el entorno de ejecución de SDV de código abierto Eclipse Velocitas). El entorno de pruebas digital.auto es de código abierto y gratuito. Puedes probarlo en [playground.digital.auto](#). ¡Esperamos tus comentarios! En segundo lugar, estamos fomentando la interoperabilidad a lo largo de la cadena de valor de los SDV.

Para impulsar un enfoque #digitalfirst, es necesario integrar a la perfección diversas herramientas y plataformas técnicas, desde las pruebas iniciales, pasando por el desarrollo virtualizado en la nube, hasta la implementación en el hardware del vehículo de destino. El sitio web [interop.digital.auto](#) resume nuestras actividades actuales en este ámbito y ofrece una serie de puntos de prueba de interoperabilidad

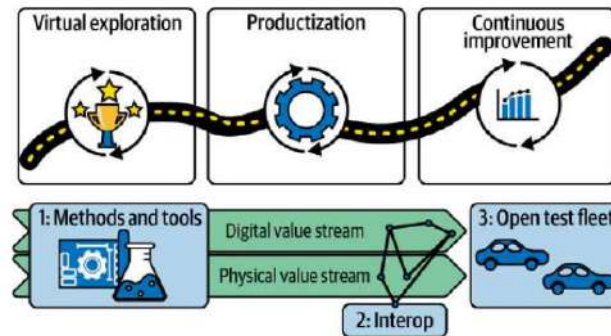


Figura 6.1: Las áreas de enfoque digital.auto

que hemos implementado en la comunidad digital.auto.

En tercer lugar, creemos que para liberar todo el potencial de los SDV, debemos crear un ecosistema dinámico de fabricantes de equipos originales (OEM), startups, desarrolladores e innovadores que colaboren para probar nuevos y emocionantes casos de uso. Esto requerirá una infraestructura de pruebas para SDV que aún no está disponible. Nuestra visión es colaborar en una flota de pruebas abierta para SDV, que incluya una tienda de aplicaciones abierta para nuevas funciones experimentales, ejecutada en un entorno seguro. Si esto le resulta interesante, visite www.digital.auto y contáctenos para que esta visión se haga realidad.

Acerca de los autores

Dirk Slama es vicepresidente de Robert Bosch GmbH y presidente de la iniciativa digital.auto. También dirige el Laboratorio de AIoT en el Instituto Ferdinand-Steinbeis, donde ocupa una cátedra. Dirk cuenta con 25 años de experiencia en proyectos de TI a gran escala en los sectores de la automoción, la fabricación, las finanzas y las telecomunicaciones. Es coautor de cuatro libros, con más de 50,000 ejemplares vendidos. Sus credenciales académicas incluyen un doctorado en sistemas de información, un MBA y un diploma en informática.

Achim Nonnenmacher impulsa innovaciones en vehículos definidos por software como directivo sénior en ETAS GmbH (una filial de Robert Bosch). Como copresidente de la iniciativa digital.auto, contribuye a acelerar la adopción de nuevas tecnologías en la industria automotriz, basada en casos prácticos. Previamente, lideró innovaciones de productos a gran escala en el sector de la movilidad, validando hipótesis empresariales y tecnológicas, así como las necesidades de los usuarios para proyectos estratégicos. Achim posee un doctorado del Instituto Suizo de Tecnología (EPFL), una maestría en física y un título ejecutivo en aceleración de la innovación de la Universidad de California en Berkeley.

Thomas Irawan es presidente de ETAS y presidente del Consejo de Administración. Antes de incorporarse a ETAS, Thomas trabajó durante 16 años en Robert Bosch GmbH, ocupando diversos puestos de liderazgo en fabricación, calidad, desarrollo e ingeniería. Fue director técnico de planta de Bosch Tailandia y vicepresidente sénior de gestión de calidad en la división de Control de Sistemas de Chasis. Recientemente, dirigió la unidad de negocio de experiencia del conductor para conducción asistida, automatizada y conectada en la división de Soluciones de Computación Interdisciplinaria. Thomas obtuvo su doctorado en física en la Universidad de Dortmund (Alemania).