CONVERGING ON A ZERO-TRUST BLUEPRINT TO CLOSE THE SECURITY AND SAFETY GAPS IN THE AUTOMOTIVE SOFTWARE INDUSTRY

Facilitating a zero-trust model of digital identity, confidential compute, and user-to-app encryption for software-defined vehicles

Table of contents

Introduction

Today's vehicles are more software than metal. High-performance computing (HPC) and in-vehicle networks have become central to all aspects of the driving experience, from body control, driver assistance, infotainment, and powertrain systems. There is a customer expectation that a vehicle's capabilities can be updated over the air with a centralized electric/electronic (E/E) architecture that will redefine today's release cycles.

Indeed, software-based features and services are projected to be a $640 billion market by 2031, growing from about 9%, or $181 billion, of OEMs' current annual revenue to 22%, according to Capgemini Research Institute.

As vehicle software content and connected services grow, so does code complexity and the need for the entire automotive ecosystem to grapple with a new set of risks. Whether a hacker takes over a driver assistance function, tampers with a software update, or exploits software vulnerabilities of keyless entry, the number of malicious attacks will only rise.

While most OEMs are only at the beginning of their software-driven transformation, less than 10% of OEMs, on average, believe that they are well prepared to implement cyber security measures, according to Capgemini Research. (See Figure 1.)

Ultimately, the base contract of future mobility solutions is safety, reliability, and robustness. As such, trust must be maintained and reinforced. All features provided by software must comply with this contract. We believe that adherence to zero-trust principles will allow automotive OEMs, software developers, and suppliers to create a strong technology foundation for data privacy, security, and cyber security requirements.

This white paper introduces a vehicle security blueprint developed by Capgemini in collaboration with Arm and NetFoundry. It provides a point of view, and proof of concept that assumes no prior trust exists between services, transactions, and software components, whether in the vehicle's hardware or offboard infrastructure.

The zero-trust blueprint proposes that automakers and suppliers prioritize addressing digital identity, the hardware-based root of trust (RoT), and explicit cryptography-based authentication and authorization of all interactions, including app-level micro-segmentation. Hardware and software developers must be granted least-privileged and just-in-time access without dependencies on the network and cloud infrastructure.

How prepared are OEMs to implement cybersecurity measures in the following area, share of OEMs

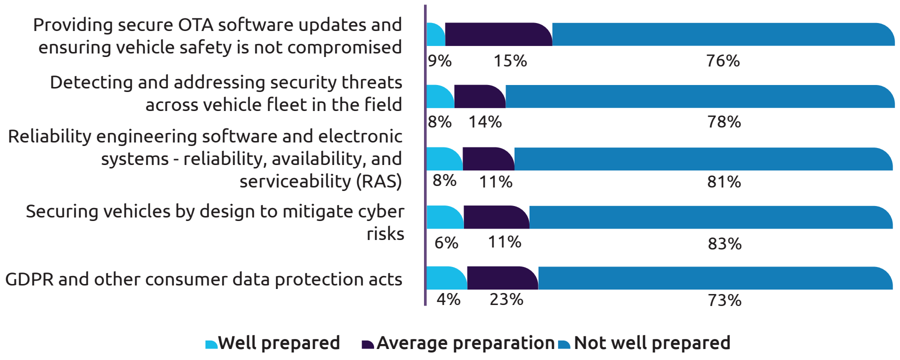


Figure 1: The majority of OEMs are not prepared to implement cybersecurity measures

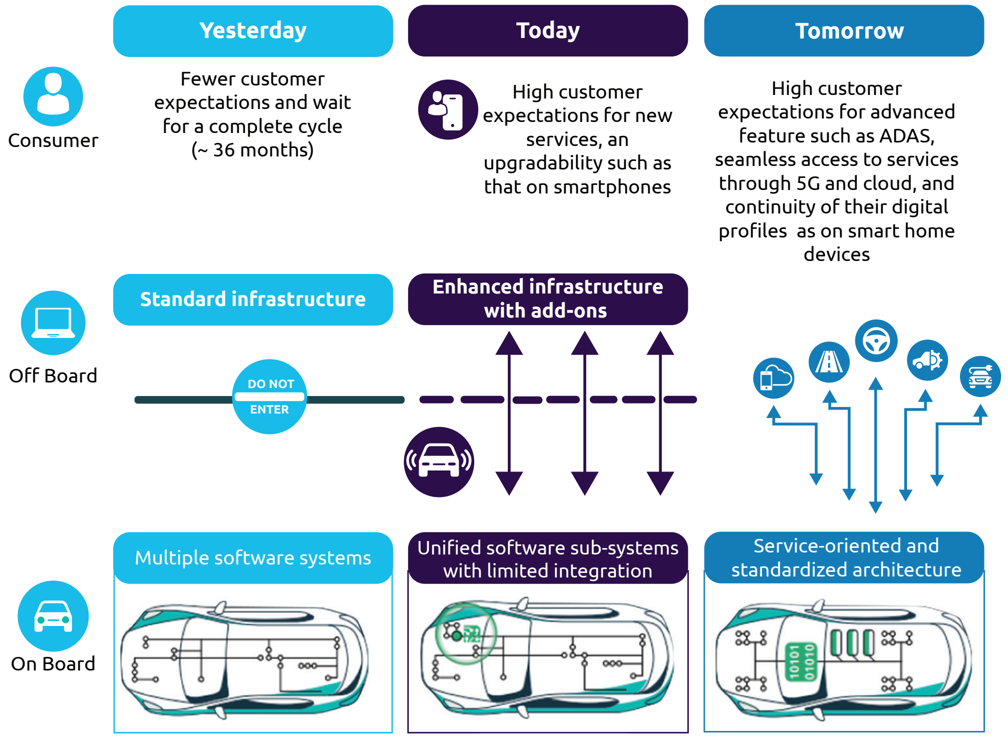
Source: Capgemini Research Institute

The software-driven transformation in automotive

A software-driven transformation is compelling automakers to rethink the way they see and build vehicles, along with how they can monetize new applications and services.

Making this possible requires significant investment by OEMs to add computing power to vehicles and disrupt the traditional E/E architecture. (See Figure 2.) Specialized ECUs combine into powerful centralized computing platforms. These platforms are service-oriented that are able to deliver a large and evolving range of services with connectivity to offboard systems. There is nothing fundamentally new about a service-oriented architecture – it has been mainstream for decades in the IT world – but now cloud-native technologies are being adopted with intensity in the automotive sector.

A standard plan-and-control approach is necessary but no longer sufficient. A more integrated adapt-and-react approach needs to be implemented.



Yesterday
Today
Tomorrow
Consumer
Fewer customer expectations and wait for a complete cycle (~ 36 months)
High customer expectations for new services, an upgradability such as that on smartphones
High customer expectations for advanced feature such as ADAS, seamless access to services through 5G and cloud, and continuity of their digital profiles as on smart home devices
Off Board
Standard infrastructure
Enhanced infrastructure with add-ons
DO NOT ENTER
On Board
Multiple software systems
Unified software sub-systems with limited integration
Service-oriented and standardized architecture

Figure 2: Evolution of the end-to-end vehicle system architecture

Source: Capgemini Research Institute

This new E/E and HPC compute architecture is an unprecedented change to the whole vehicle development process that now puts software at the epicenter of service innovation. In turn, there will be a substantial opportunity for cyberattacks and unintended intrusion into a target-rich environment. From vehicle safety and control systems to supply chain data in fleet vehicles, infotainment, and sensitive personal and financial data, all deserve proportionate risk management. (See Figures 3.)

Without hesitation, the automotive industry will need to defend a larger attack surface because of connected vehicles and business models where customers are expecting over-the-air (OTA) software updates. For example:

- Frontier risks from as yet to be defined mobility experiences from a network of charging stations, ride-sharing services and smart city infrastructure
- Well-known privacy concerns because car data is collected, shared, and transmitted, including location, driving behavior, and account information
- New risks from digital continuity that is being established between an automaker's information, operational, and production technologies within an increasingly complex ecosystem

Property of Security	Example Issues	Key Assessment Questions
Availability	• Ransomware • Denial of service	• What are the ECUs that need protection? • Will this impact vehicle use and safety?
Confidentiality	• Open connection(s) • Data theft • Personally identifiable information leakage	• How do you manage connections efficiently and securely? • Who owns the data security? • How is data collected and stored?
Integrity	• Trusted logs • App runtime integrity • Secure boot	• How long is the data retained? • Who/what needs access to the data? • How can we know where the data came from?

Figure 3: Key security issues facing the software-driven transformation in automotive

Source: Capgemini

Zero-trust blueprint: The key components

This zero-trust blueprint is rooted in the principle of "never trust, always verify" to minimize the risk of unwanted intrusions and advanced persistent threats. Zero trust is a set of principles to protect modern digital environments through strong identity management, app-level micro-segmentation, blocking inbound ports and applying least privileged access control, regardless of underlying networks and cloud providers.

In an automotive context, zero-trust begins by establishing the vehicle as an entity with its own networks and nodes that benefit from robust identity management and micro-segmentation. It may happen that part of the software will not be located in the vehicle but in the network or the cloud. So functionally speaking, the physical vehicle does not correspond anymore to the mobility experience.

When converging on a zero-trust blueprint for automotive, we started by combining the principles of confidential computing and confidential connectivity:

- Confidential computing secures data in use by isolating it in a hardware-based trusted execution environment (TEE). These secure and isolated environments prevent unauthorized access or modification of applications and data while they are in use, thereby increasing the security level of organizations that manage sensitive and regulated data. While data is being processed, it is invisible and unknowable to the operating system, the hypervisor of a virtual machine, and other compute-stack resources. It also can't be seen by cloud providers or their employees.

- Confidential connectivity secures the network with identity-based connections to create application-specific and highly secure private zero-trust connections to software services hosted by service and cloud providers. Connections are identity-based. Identity is initiated and established with vehicle-specific Public Key Infrastructure (PKI) that is based on hardware security modules. Connections and services are visible only to the entities authorized to use them. The network is dark to all others. In short, these systems are invisible and unknowable to the outside world, making them impervious to cyberattacks.

We initially outlined ten requirements to be applied in an automotive context:

- **Secure product development** - Define the requirements, architectures, and implementation throughout the entire systems lifecycle. Then check all of the work and document each step. Security is integrated and repeated at each phase

- **Defense-in-depth strategy** - Vehicles with security controls that are applied at different levels of hardware and software. Robust baseline assurances on each ECU

- **Secure digital identity** - Foundation for authentication and authorization before any connection is made. Access is policy-based and dynamically enforced. Services are mapped at the vehicle level

- **Secure provisioning** - Leverages the digital ID as a key component for onboarding the connected services and nodes within the vehicle. Ensures configuration management is performed precisely as intended

- **Confidential computing** – Private, trusted compute via hardware. Onboard secure software, sandboxed application content, discrete cryptographic functions, and confidential data "in-use" are all possible applications

- **Secure over-the-air (OTA) updates** – Deploy patches for existing applications, new features, and fix flaws. Micro-segmented, private connectivity ensures the authenticity of software and applications

- **Data encryption** - End-to-end encryption of data within the vehicle, in-transit to and stored within the IT infrastructure, for appropriate handling by the final user. Managed authentication and authorization to sensitive features, data, and resources
- **Micro-segmentation and least privilege** - Provide just-in-time and just enough access per application. Prevents lateral movement and advanced persistent threats
- **Assume system compromise** - Apply threat management at the system, network, and application levels. Log all inconsistencies, software changes, and out-of-bounds hardware interactions
- **Actionable Information** - Using log and application data to create dynamic, enforceable policies that align with security goals ensures resilience throughout the entire lifecycle of the vehicle

We believe this blueprint will minimize the attack surface for connected vehicles by changing the security posture from reactive and bolted-on to secure by design with zero trust built-in. It supports connected ECU-based vehicle architectures, HPC platforms, containers and microservices, and vehicle-to-everything (V2X) services.

Automotive OEMs will need to go well beyond secure connectivity to address issues and considerations inside the vehicle from the ECUs, networks, and system components. At the same time, achieving zero trust is not straightforward due to several factors, including constraints of traditional vehicular communication protocols and priority given to functional requirements and cost controls.

For example, the performance characteristics of existing ECUs may not be sufficient for cryptographic operations and materials management. Additionally, there is an inherent trust that is given to different components within a car. This has led to some 'mission-critical' systems communicating in the clear on closed networks, which are susceptible to physical manipulation and modification. With the complexity of onboard systems, there will be a knock-off effect of safety issues that increasingly will be software-related. While somewhat easily mitigated, implementing security checks on these communications can also have a negative impact on the safety-rated timings of these systems.

Proof of concept example: Secure delivery of software in a Cellular-V2X/ telematics scenario

To bring this blueprint to life, we applied several requirements to demonstrate the secure connectivity of a future telematics scenario. In this use case, a passenger or commercial vehicle shares telematics data, such as GPS location, while staying in touch with an OEM fleet operator's control tower.

We assume that a vehicle's hardware and software are decoupled in a service-oriented architecture. A key objective is to ensure that software deployed on the vehicle securely communicates with offboard cloud services. We also want to make sure that OEMs can provision applications and services dynamically on the vehicle.

In this proof of concept, we focus on achieving zero-trust principles through the following capabilities:

- **Vehicle enrollment**: Association of a vehicle identity (public/private keys provisioned by the onboard HSM) with the customer profile. The Arm Parsec Service, a Cloud Native Computing Foundation (CNCF) project, provides abstraction to the crypto functions across different providers

- **Secure software delivery**: Secure delivery of application and platform containers into the vehicle that will be hosted within the vehicle's container cluster runtime. Capgemini automotive software foundation for dynamic and containerized environments ("Autobits") allows vehicle functions to be deployed and orchestrated as containers that can coexist with ECUs for safety-critical applications

- **Zero-trust connectivity**: Secure communication to end services with end-to-end encryption and obfuscation of connectivity between the vehicle and services. The NetFoundry zero-trust platform is used to provide secure, outbound only, private communication with zero-trust networking between sensors and containers operating inside the car on virtualized compute infrastructure, or inside the applications themselves, and outside to cloud infrastructure for a number of business software applications including business tools, ML inferencing, and AWS IoT core, etc

- **Services entitlement**: Association of vehicles to the application services allows OEMs to provision and deploy new services to the users and manage the QoS of the provisioned services

- **Network and cloud resilience**: Applications running on vehicles need highly resilient network connectivity to achieve reliable connectivity. Vehicles will use multiple network connectivity technology, including private, dynamic, self-healing mesh provided by NetFoundry. The goal is to minimize out-of-coverage geographical regions

To be clear, in our proof of concept, we focused on secure connectivity and the establishment of hardware root of trust, which is necessary but insufficient given the additional requirements of the blueprint. In turn, we applied the following technical implementation and architecture. (See Figure 4.)

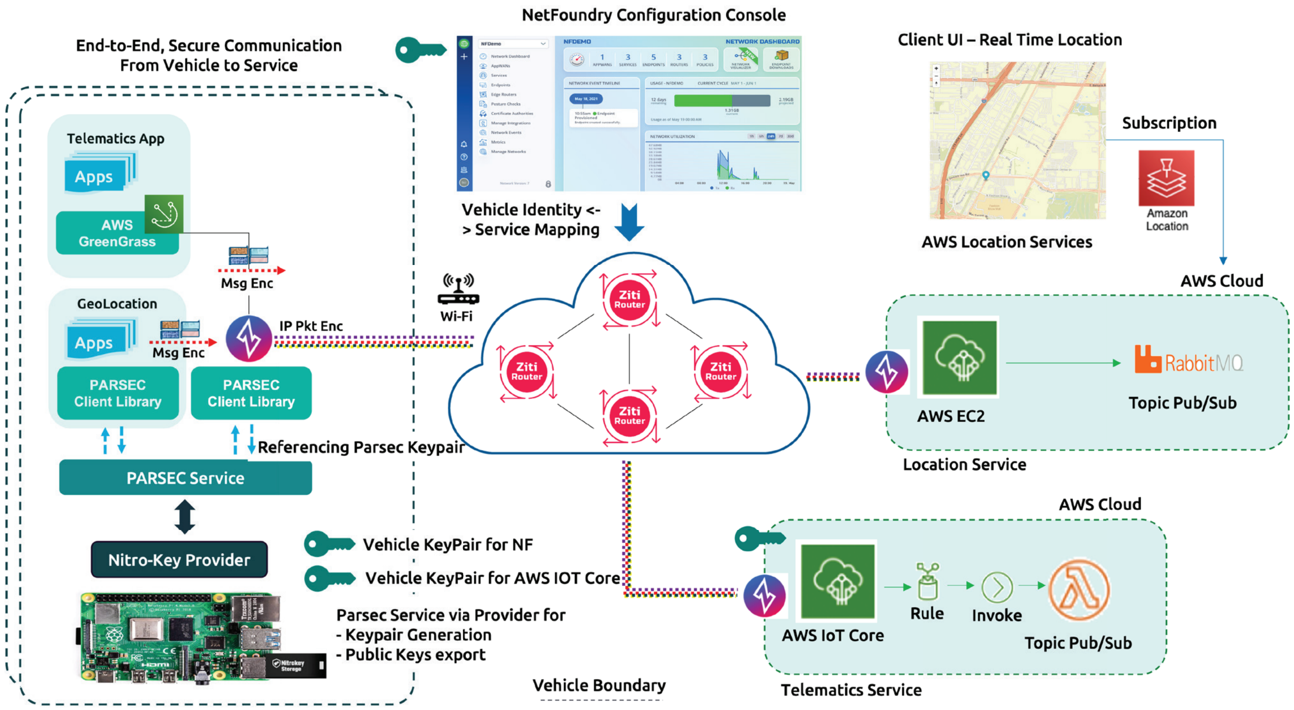


Figure 4: The technical architecture of the proof of concept

Source: Arm, Capgemini, and NetFoundry

The technical details of the implementation include:

Establishing a secure connection

Our proof of concept implements secure private application wide area networks (AppWANs) to connect applications running in vehicles to resources hosted in public or private clouds, other data centers, or edge locations. We implemented specific private connections to Amazon Web Services and notably AWS IoT Core.

All application-level connections and those at the infrastructure level are secured so that developers can choose fine-grained control per application versus system-wide connectivity for all containers and virtual machines on the vehicle compute environment.

NetFoundry automation creates and deploys networks, including data plane, control plane, and management functions, necessary to operate these systems as a service. Endpoints are created by embedding network connectivity into applications using open-source SDKs (OpenZiti) or by implementing software-defined tunnelers which run on the operating system. For example, tunnelers are used to connect to AWS instances hosting AWS IoT Core. Whether SDK or tunneler, these endpoints can even be run in the TEE of an Arm-based device, i.e., a confidential network inside confidential compute.

The endpoints open an outbound-only session to the fabric, which only listens to authenticated and authorized endpoints based on the embedded identity.

This approach provides the data transport across the zero-trust network overlays. Policies define specific edge routers for specific endpoints and how they are used for network transport segregation and optimization. AppWANs are implemented to define the specific services that can be accessed by one or more client endpoints. Endpoints can access only the specified services they are authorized to connect to. Everything else is invisible.

Securing containers of applications and services

We deployed the Capgemini Autobits software framework to provide automated secure deployment and orchestration of container applications on the vehicle edge that make use of the Parsec SDKs and NetFoundry services for secure communication to cloud services.

Using cryptographic APIs for applications

We followed the zero-trust blueprint's security by using Parsec, an open-source project, to provide a common set of APIs to manage security services in the hardware root-of-trust mechanisms (HSM/TPM). (See Figure 5.) Parsec can broker access to the root of trust for various applications such as NetFoundry and AWS IoT core to provide a secure multi-tenancy environment. This approach greatly simplifies the security interface as Parsec deals with the intricacies of the various hardware roots of trust with pluggable provider modules. On the front end of Parsec are client interfaces available for multiple coding languages that expose the common set of APIs. This creates an abstraction layer necessary to maintain the portability of code while enabling the highest level of security available in the system.

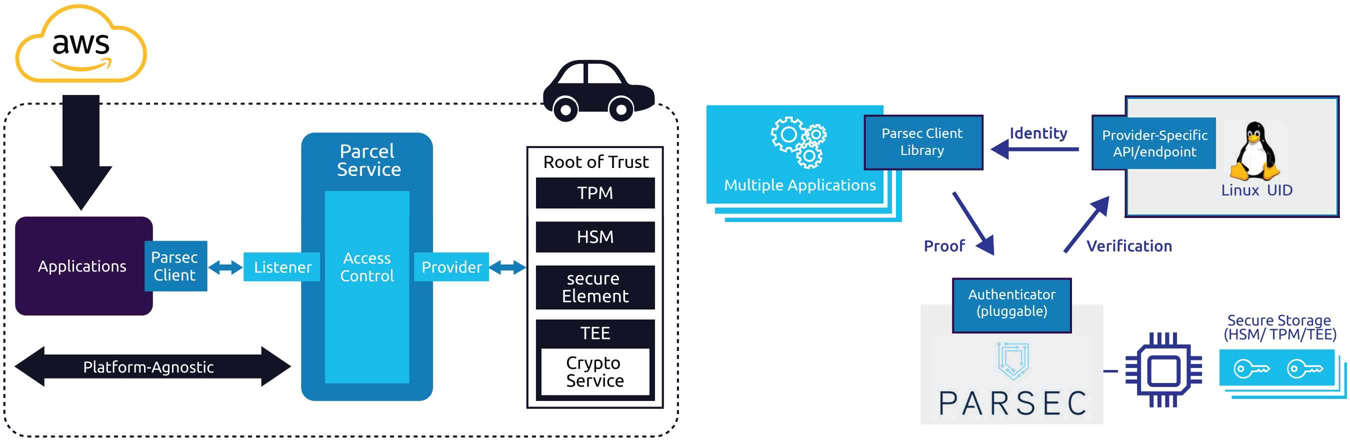


Figure 5: Portable hardware root of trust
Source: Arm

Summary

The zero-trust blueprint developed by Capgemini, Arm, and NetFoundry provides a next-generation foundation to accelerate innovation in software and the transition to new digital business models in the automotive industry. By design, it solves the key challenges and addresses critical vulnerabilities in securing connectivity from the edge and cloud to in-vehicle systems and applications. The blueprint abstracts out infrastructure complexity and brings cloud-native agility, scalability, and seamless mobile connectivity to connected vehicles with a standards-based design and open-source components.

The zero-trust blueprint helps automotive OEMs and their ecosystems continue their focus on developing autonomous driving while implementing secure, scalable systems to monetize applications and new connected vehicle services. Trust and security will pave the way for the future of mobility.

Acknowledgments

Capgemini Group

- Jitendra Thethi, Vice President, Software-defined Systems
- Joby Jester, Director, Embedded Cybersecurity
- Anantha Prakash Tungarakodi, Director, Open Hardware
- Walid Negm, Chief Research & Innovation Officer
- Hitesh Pandya, Global Lead Future of Mobility
- Dr. Pascal Feillard, Director Automotive Transformation

Arm

- Marc Meunier, Senior Manager, SW Ecosystem Development
- Oliver Lythgoe, Senior Marketing Manager, Automotive
- Angela Williams, Director, Channel Sales
- Reed Hinkel, Senior Solutions Architect

Netfoundry

- Phillip Griffiths, Vice President, Global Business Development and Alliances
- Brent Doncaster, Senior Director, Marketing
- Sathish Shivarudrappa, Senior Solution Architect

Appendix

AWS IoT Greengrass

AWS IoT Greengrass is an open-source edge runtime and cloud service for building, deploying, and managing device software. AWS IoT Greengrass provides pre-built components for common use cases so you can discover and import, configure, and deploy applications and services at the edge without the need to understand different device protocols, manage credentials, or interact with external APIs. You can also create your own components or simply re-use common business logic from one AWS IoT Greengrass device to another.

Capgemini Autobits

Capgemini Autobits is a collection of software frameworks that are being collaboratively developed with the automotive ecosystem to address the service-oriented shift to zonal and high-performance compute (HPC) architectures such as vehicle edge hardware management, mixed-criticality hypervisors, secure software-over-the-air updates, orchestrated containers, and zero-trust principles.

Confidential Computing

The Confidential Computing Consortium (CCC) brings together hardware vendors, cloud providers, and software developers to accelerate the adoption of Trusted Execution Environment (TEE) technologies and standards. CCC is a project community at the Linux Foundation dedicated to defining and accelerating the adoption of confidential computing. It will embody open governance and open collaboration that has aided the success of similarly ambitious efforts. Confidential Computing protects data in use by performing computation in a hardware-based TEE.

Open Ziti

Easily embed secure, high-performance networking into your apps and solutions without managing networking. Ziti is the leading open-source networking platform, enabling zero-trust networking from anywhere to anywhere, over the internet. NetFoundry's industry-leading network-as-a-service (NaaS) solutions are built on Ziti, so you can leverage quick-start solutions from NetFoundry, including the NetFoundry Ziti Fabric (from any internet connection), to instantly connect edge, cloud, and B2B applications.

Parsec

A key part of Project Cassini's security pillar, Parsec, which is compliant with PSA Certified, enables software developers to maintain portability while taking advantage of best-in-class security features that can be applied directly to their applications. Conceptualized at Arm, Parsec is an official Cloud Native Computing Foundation sandbox project, which has now been embraced by the open-source community and integrated with operating systems including Fedora, OpenSuse and Yocto Linux.

Project Casinni

As the nature of compute changes, the edge plays an increasingly crucial role in supporting diverse systems with a range of power and performance requirements. To deliver on service level agreements at scale for enterprises, the edge must embrace cloud-native software principles.

Project Cassini is the open, collaborative, standards-based initiative to deliver a seamless cloud-native software experience for devices based on Arm Cortex-A. Developers of IoT and infrastructure edge solutions can access the power of Cassini today with Arm SystemReady, PSA-certified silicon and development boards, and OS Linux support from the Arm ecosystem.

SOAFEE

The aim of the SOAFEE project is to bring the benefits of a cloud-native development environment to address the specific challenges and constraints of the automotive domain, such as functional safety (FuSa) and fast and precise real-time control.

The SOAFEE project leverages a cloud-native framework in order to benefit from the best practices and standards to support the ability to deploy workloads to heterogeneous compute architectures with a mixture of application processors and real-time processors with an array of accelerators available. The SOAFEE project brought together automakers and semiconductor and cloud-technology leaders to define a new open-standards-based architecture for the software-defined vehicle.

Capgemini engineering

About Capgemini Engineering

Capgemini Engineering combines, under one brand, a unique set of strengths from across the Capgemini Group: the world leading engineering and R&D services of Altran – acquired by Capgemini in 2020 - and Capgemini's digital manufacturing expertize. With broad industry knowledge and cutting-edge technologies in digital and software, Capgemini Engineering supports the convergence of the physical and digital worlds. We help clients unleash the potential of R&D, a key component of accelerating their journey towards Intelligent Industry. Capgemini Engineering has more than 52,000 engineer and scientist team members in over 30 countries across sectors including aeronautics, space and defense, automotive, railway, communications, energy, life sciences, semiconductors, software, and internet and consumer products.

About Arm

Arm technology is at the heart of a computing and data revolution that is transforming the way people live, and businesses operate. Our energy-efficient processor designs and software platforms have enabled advanced computing in more than 200 billion chips, and our technologies securely power products from the sensor to the smartphone and the supercomputer. Together with 1,000+ technology partners, we are at the forefront of designing, securing and managing all areas of AI-enhanced connected compute from the chip to the cloud. www.arm.com

About NetFoundry

NetFoundry is the leader in Cloud-Native Networking, enabling businesses to simply, securely, and cost-effectively connect distributed applications across edges, clouds and service meshes.

The NetFoundry platform, delivered as a service, enables businesses to embed zero-trust security into applications and connect applications to users without the costs and complexity of VPNs, custom hardware, and private circuits. NetFoundry's platform is accessed via APIs, SDKs, and DevOps tools integrations, enabling practitioners, application developers, and network administrators to get the levels of automation and agility which are only possible with connectivity-as-code. www.netfoundry.io

All information in this document is provided "as is" and without warranty or representation. This document may be shared freely, attributed and unmodified.

For more information please visit:

www.capgemini-engineering.com

Contact us at:

zerotrustblueprint@capgemini.com