PERSEUS

Be Simple, Be Secure

Corporate & Solution Overview

Foundational System Software & Secure Virtualization for Software-Defined Vehicles *(and Beyond)*

Perseus is an automotive system software company developing foundational system software for Software-Defined Vehicle (SDV) architectures.

With more than **18 years of experience embedded systems and virtualization**, Perseus provides system-level software that helps OEMs and suppliers scale in-vehicle computing while meeting real-world efficiency, security, and performance requirements.

Perseus' portfolio includes an ISO 26262 ASIL-D certified automotive hypervisor for CPU and MCU platforms, along with complementary security technologies and fast-boot solutions for the SDV era.

www.cyberperseus.com
sales@cyberperseus.com

INDEX

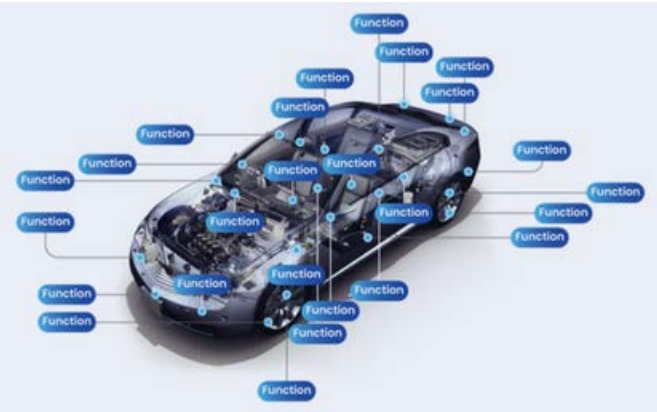
INDUSTRY BACKDROP

As vehicles transition toward Software-Defined Vehicle (SDV) architectures, the number of software functions, workloads, and interactions continues to increase. Historically, this growth has been managed by adding dedicated Electronic Control Units (ECUs) and System-on-Chips (SoCs), resulting in fragmented architectures that are complex, costly to integrate, and difficult to evolve over a vehicle's lifecycle.

Under prevailing design approaches, electric vehicles can require up to 1,000 SoCs, while autonomous vehicles may require 2,000 or more. This is largely driven by the physical separation of safety-critical and non-critical functions across dedicated hardware, increasing system complexity, integration effort, cost, and cybersecurity exposure.

The Need for Consolidation

SDV principles aim to reverse this trajectory. While software complexity continues to grow, SDV architectures enable hardware consolidation by allowing multiple operating systems and mixed-criticality workloads to safely coexist on shared compute platforms.



This transition is still emerging, and many OEM programs have not yet fully adopted SDV design methods in mass production. As a result, a gap remains between SDV ambitions and production-ready system architectures capable of supporting scalable, centralized vehicle computing.

The Shift to Software-Defined and Centralized Vehicle Architectures

The automotive industry is transitioning from hardware-centric vehicle design toward Software-Defined Vehicle (SDV) architectures, where software has become the primary driver of functionality, differentiation, and long-term value.

As vehicles grow more connected, automated, and updateable over their lifecycle, multiple operating systems and mixed-criticality workloads must coexist within constrained cost, power, and safety boundaries.

Historically, these demands were addressed by adding dedicated ECUs for individual functions, resulting in fragmented architectures with rising hardware count, integration complexity, and cost. As of 2026, this approach is no longer sustainable.

AUTOMOTIVE VIRTUALIZATION AND THE TRANSITION TO SOFTWARE-DEFINED VEHICLES

The Role of Virtualization and Automotive Hypervisors

To address these challenges, the industry is moving toward centralized and zonal computing architectures, where multiple functions execute on shared CPUs and MCUs. Foundational system software has emerged as a core architectural enabler of this transition, supporting scalable, production- ready SDV platforms and long vehicle lifecycles.

OEMs and automotive suppliers are adopting foundational system software based on virtualization. Automotive hypervisors enable multiple operating systems and applications to safely coexist on shared hardware while remaining isolated and deterministic.

Did you know, Global OEMs like BMW and Hyundai have already committed to adopting hypervisors?

By allowing workloads with different safety, security, and real-time requirements to run concurrently on shared CPUs and MCUs, hypervisors improve hardware utilization, simplify system architectures, and support scalable SDV platforms, while preserving efficiency, security, and performance at the system level.

Key Trends Shaping the Automotive Hypervisor Landscape

Software-Defined Vehicle Architectures Become the Default

Automakers are moving from hardware-defined vehicle platforms toward Software-Defined Vehicle (SDV) architectures, where functionality is introduced, updated, and differentiated through software over long lifecycles. Hypervisors are becoming foundational system software in these platforms, enabling multiple operating systems and workloads to run on shared CPUs and MCUs with strong isolation and predictable behavior. This shift improves hardware utilization, reduces reliance on dedicated ECUs, and supports centralized and zonal E/E architectures suitable for production deployment.

Rising Compute Density Driven by ADAS and Automation

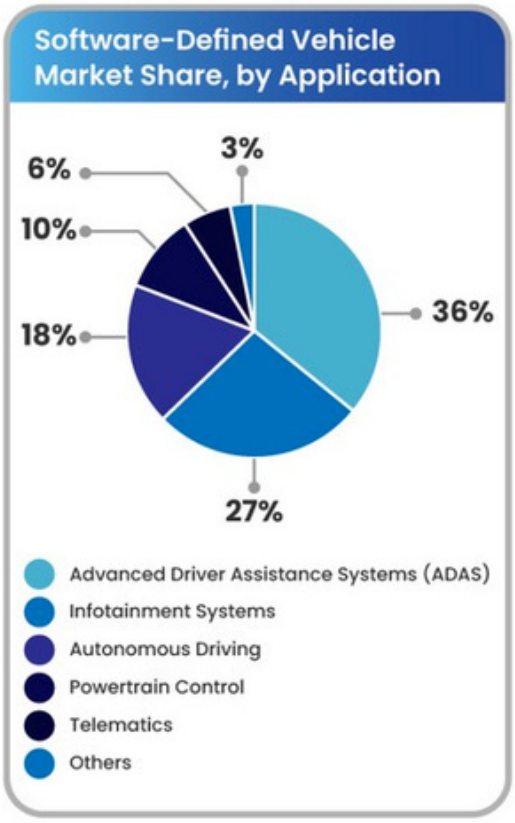
Advanced Driver-Assistance Systems (ADAS) and higher levels of automation significantly increase in-vehicle compute density. Sensor fusion, perception, and control workloads must execute alongside infotainment and connectivity software on constrained hardware. Hypervisors provide the isolation, scheduling, and deterministic resource control required to ensure safety-critical workloads meet real-time constraints without interference from non-critical domains.

Functional Safety Moves to the System Architecture Level

As multiple mixed-criticality workloads are consolidated onto shared compute platforms, functional safety can no longer be addressed at the ECU or operating system level alone.
Automotive hypervisors are increasingly designed and certified to meet standards such as ISO 26262 and ASIL, ensuring predictable timing, controlled resource allocation, and fault containment across the entire system. Safety-certified hypervisors are now a prerequisite for centralized and zonal SDV platforms entering mass production.

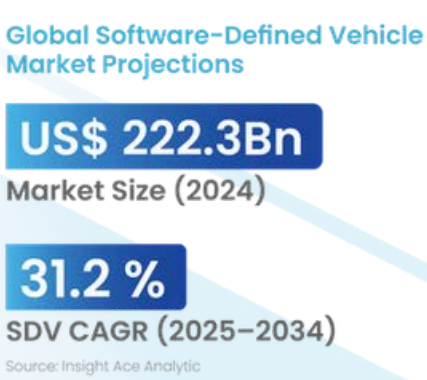
System-Level Cybersecurity Becomes Essential Infrastructure

Expanded vehicle connectivity, OTA updates, and software-driven feature growth increase the attack surface of modern vehicles. Hypervisors are increasingly used to enforce system-level isolation below the operating system layer, limiting fault propagation and containing security incidents. By controlling access to hardware resources and inter-domain communication at the lowest levels of the stack, secure hypervisors strengthen the security posture of connected and automated vehicle platforms.



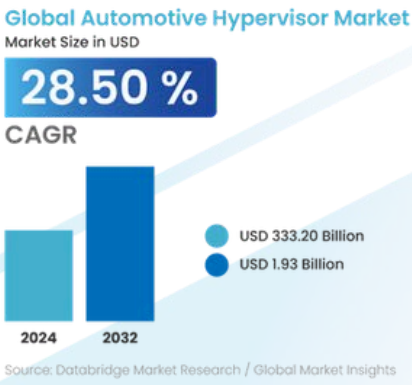
Acceleration of Development and Validation Through Virtualization

Virtualized system architectures are increasingly used not only in production vehicles but also in development, testing, and validation workflows. Hypervisors enable virtual bring-up, early integration, and system-level testing before physical hardware is available, reducing development time and cost. This trend is becoming especially important as SDV programs grow in complexity and certification timelines become a critical bottleneck.



Increasing interest in SDVs is driving a rapid

increase in the need for hypervisors



Benefits of Automotive Hypervisors

Automotive hypervisors form the foundation of system software for Software-Defined Vehicles (SDVs).
By virtualizing hardware resources and enforcing system-level control, they enable efficient use of shared compute, strong isolation and security between workloads, and deterministic, real-time performance - supporting scalable, production-ready SDV architectures.

Efficiency - Hypervisors improve system efficiency by enabling safe, predictable sharing of hardware across workloads.

- **Resource & Hardware Consolidation** – Multiple operating systems and workloads execute on shared CPUs and MCUs, reducing reliance on dedicated ECUs and improving utilization of compute, memory, and I/O.
- **Legacy Software Reuse** – Existing AUTOSAR, RTOS, Linux, and legacy applications can run unchanged in virtualized environments, reducing redevelopment effort.
- **System-Level Cost & Power Efficiency** – Fewer SoCs and ECUs lower semiconductor, validation, maintenance, and energy costs over long vehicle lifecycles.

Security - Hypervisors enforce security at the system architecture level through isolation and controlled interaction.

- **Strong isolation of Mixed-Criticality Workloads** – Safety-critical and non-critical systems execute in isolated environments, limiting fault propagation and attack impact.
- **Controlled Access & Communication** – Privilege and communication policies restrict unauthorized interactions between software domains.
- **Secure OTA Enablement** – Software updates are isolated from safety-critical execution domains, reducing risk during deployment.

Performance - Hypervisors enable predictable, real-time behavior across mixed workloads.

- **Deterministic Execution** – Time and safety-critical workloads maintain predictable timing even when sharing hardware with non-critical functions.
- **Workload Prioritization** – Compute resources are scheduled and allocated based on criticality and real-time requirements.
- **Multi-OS Support** – Real-time and general-purpose operating systems (e.g. AUTOSAR, RTOS, Linux) coexist safely on shared hardware.

Lifecycle Scalability - Across efficiency, security, and performance, hypervisors support long-term platform evolution.

- **Software-Driven Expansion** – New features and capabilities can be introduced through software without hardware redesign.
- **Reduced Architectural Complexity Over Time** – Centralized workload management simplifies integration, testing, and maintenance as SDV platforms evolve.

INTRODUCING PERSEUS FOUNDATIONAL SOFTWARE FOR SDV PLATFORMS

Perseus develops foundational system software that operates at the bare-metal and system layer, addressing efficiency, security, and performance at the lowest levels of the software stack to enable scalable, Software-Defined Vehicle architectures.

Its solutions provide virtualization, strong isolation, deterministic resource control, and fast system startup - capabilities that become critical as vehicle computing consolidates onto shared CPUs and MCUs.

Perseus enables multiple operating systems and mixed-criticality workloads - including AUTOSAR Classic, RTOSs, Linux, Android, QNX, and legacy systems - to coexist safely on shared, multi-core hardware. By enforcing predictable execution below the operating system layer, Perseus supports hardware consolidation, simplifies E/E architectures, and enables production-ready SDV platforms aligned with centralized and zonal vehicle designs.

A Long History of Hypervisor Innovation

Founded in 2016, Perseus is led by Sangbum Suh, Ph.D., a long-time contributor to embedded virtualization and the founder and maintainer of Secure Xen ARM, the leading open-source embedded hypervisor project established in 2007. His work played a formative role in adapting virtualization concepts for resource-constrained, safety-critical embedded systems.

Perseus was founded in Seoul, Korea in 2016 by Sang-bum Suh PhD, Founder of Secure Xen Open Source Community (2007)

Perseus' leadership brings more than two decades of experience spanning academic research (including the University of Cambridge), foundational open-source system software projects, and large-scale commercial development at global electronics and semiconductor companies such as Samsung Electronics.

This combination of open-source leadership, academic rigor, and production experience underpins Perseus' ability to translate advanced virtualization concepts into certified, production-ready automotive system software - including the ISO 26262 ASIL-D certified PEGASUS hypervisor for both CPU and MCU architectures.

PERSEUS' SUITE OF FOUNDATIONAL SOFTWARE FOR SDV PROGRAMS (SHORT INTROS)

Perseus' solution portfolio provides complementary capabilities across the embedded vehicle technology architecture, delivering benefits that extend across multiple system domains rather than isolated functions.

PEGASUS is Perseus' flagship Type-1 automotive hypervisor, enabling the safe virtualization and isolation of multiple operating systems on shared hardware across CPU and MCU architectures. PEGASUS is ISO 26262 ASIL-D certified, meeting the highest functional safety requirements.

TACHYON is a Linux Fast Boot solution, reducing startup time to under two seconds while preserving the Linux ecosystem for time-critical vehicle functions.

AEGIS is a secure hypervisor solution that enforces system-level isolation and access control, protecting safety-critical workloads from threats that cannot be fully mitigated at the OS layer.

AEGIS-let is a lightweight, TrustZone-based secure kernel designed to protect ECU-level security-critical functions in resource-constrained environments.

GAIA is an MCU secure boot loader, supporting hypervisor-based and multi-OS systems by standardizing boot, initialization, and secure startup.

PEGASUS is complemented with the PEGASUS SDK, enabling teams to integrate PEGASUS into their SDV design efficiently and with confidence. The SDK comprises two components:

Workbench is the VM configuration and system build environment for PEGASUS-based platforms. It is designed to reduce the complexity of building, validating, and maintaining virtualization-based automotive systems.

User Dashboard provides runtime visibility and operational control at the hypervisor layer. It enables engineers to monitor system status, observe resource usage, and manage virtual machine lifecycles in real time.

PERSEUS' SUITE OF FOUNDATIONAL SYSTEM SOFTWARE AND SECURE VIRTUALIZATION FOR SDV PROGRAMS

DETAILED INTRODUCTIONS

To learn more, contact a member of our sales team today

sales@cyberperseus.com

PEGASUS

ISO 26262 ASIL-D Certified (CPU & MCU architectures) Addressing the Growing Complexity of Automotive Electronics

Modern vehicles increasingly depend on highly complex E/E (electrical and electronic) and software architectures. Over successive vehicle generations, new functions have typically been added by deploying additional Electronic Control Units (ECUs) and compute domains.

While effective in the short term, this approach has resulted in fragmented systems that are increasingly difficult and expensive to develop, integrate, validate, secure, and maintain—placing growing pressure on efficiency, security, and performance at the system architecture level.

As the industry transitions toward Software-Defined Vehicle (SDV) architectures, continued reliance on function-specific ECUs is no longer sustainable. Connected and automated vehicles must simultaneously support infotainment, ADAS, body, powertrain, connectivity, and gateway functions— each with different safety levels, real-time requirements, update cadences, and lifecycle constraints.

This convergence introduces challenges

- **System architecture design**
- **Mixed-criticality integration**
- **Functional safety compliance**
- **Deterministic performance**
- **Cybersecurity and fault containment**
- **Long-term scalability and cost control**

Hypervisors address these challenges by enabling virtualization and strict isolation of multiple workloads on shared hardware, reducing reliance on dedicated ECUs while preserving safety and predictability. In this context, PEGASUS serves as foundational system software for SDV platforms, providing a scalable, secure, and high-performance virtualization layer designed specifically for automotive production systems.

Introducing PEGASUS

PEGASUS is Perseus' flagship automotive hypervisor, purpose-built for safety-critical and mixed-criticality vehicle platforms.

PEGASUS is a lightweight, Type-1 (bare-metal) hypervisor, running directly on hardware without a host operating system. This architecture minimizes software layers, reduces attack surface, and enables deterministic system behavior—key requirements for production automotive systems.

PEGASUS enables the safe virtualization and isolation of multiple operating systems and workloads on shared, multi-core hardware, supporting both application-class CPUs (e.g. ARM Cortex-A) and real-time MCU-class architectures (e.g. Infineon Technologies and ARM Cortex-R52). It supports heterogeneous environments including AUTOSAR Classic, RTOSs, Linux, Android, QNX, and legacy systems, allowing them to coexist while preserving determinism, safety, and performance.

This dual-architecture support enables PEGASUS to span centralized compute, zonal controllers, and real-time control domains within a unified virtualization framework.

As of 2025, PEGASUS is ISO 26262 ASIL-D certified for both CPU and MCU architectures, meeting the highest automotive functional safety requirements and enabling deployment in the most demanding vehicle domains.

PEGASUS is not a general-purpose virtualization layer adapted for automotive use. It is purpose-built system software designed to sit at the foundation of SDV platforms - enabling consolidation, safety, scalability, and long-term maintainability across both CPU- and MCU-based automotive domains.

The PEGASUS Advantage

In the past, virtualization approaches - particularly those adapted from enterprise or desktop environments - relied heavily on instruction emulation or extensive paravirtualization. These techniques often require guest OS modification, introduce non- deterministic behavior, and increase integration and validation complexity.

PEGASUS is architected specifically for embedded and automotive systems and avoids these limitations by:

- Leveraging hardware virtualization and protection mechanisms directly
- Minimizing instruction trapping and emulation paths
- Avoiding mandatory guest OS modification
- Maintaining static, design-time configuration models suited to safety certification

By operating as a true bare-metal hypervisor, PEGASUS delivers:

- Near-native execution performance
- Predictable timing behavior
- Deterministic resource allocation
- Simplified safety analysis and system validation

PEGASUS' static configuration model and deterministic scheduling also support **system-level real-time analysis**, enabling timing validation and safety assessment across mixed-criticality domains. In addition, PEGASUS' supports configurable system architectures with explicitly defined runtime behavior.

This makes PEGASUS suitable not only for development and prototyping, but for **production SDV platforms with long lifecycles and regulatory constraints**.

Key Capabilities of PEGASUS

Full Virtualization and Isolation - PEGASUS enables multiple virtual machines to operate independently on shared hardware. Each VM executes within a strictly defined resource sandbox, ensuring that faults, or failures in one domain do not propagate to others. This is foundational for mixed-criticality automotive systems.

Rust-Based System Implementation – PEGASUS is written in Rust, leveraging compile-time enforcement of memory safety and data-race freedom through Rust's ownership and concurrency model. This removes entire categories of runtime defects typical in C/C++ systems code, without introducing garbage collection or unpredictable runtime overhead, making it well suited for embedded automotive
platforms.

System-Level Privilege Controls - PEGASUS enforces privilege separation, controlled inter-VM communication, and hardware-backed isolation at the hypervisor layer. Privilege controls are applied below the guest operating systems, limiting the blast radius of misbehaving software and supporting defense-in-depth architectures.

VirtIO-Based IO Device Virtualization - PEGASUS uses VirtIO-based para-virtualized IO devices to enable efficient sharing of peripherals such as networking, storage, and consoles. This approach significantly reduces virtualization overhead compared to full device emulation while enabling driver reuse across heterogeneous guest OS environments.

Deterministic Resource Management - PEGASUS provides deterministic scheduling and resource allocation for mixed-criticality workloads. Virtual CPUs can be statically bound to physical cores, and memory and interrupt access are strictly controlled, enabling predictable real-time behavior for safety-critical domains alongside less critical workloads.

Application of PEGASUS in Automotive Systems

Centralized and Zonal Computing - Consolidation of multiple vehicle domains onto centralized or zonal compute units while maintaining isolation.

High-Performance Computing (HPC) - Coexistence of ADAS, infotainment, gateway, and connectivity workloads on shared high- performance SoCs.

MCU-Centric Real-Time Systems - Powertrain, chassis, and body control functions requiring deterministic execution and ASIL-D-level safety.

ADAS Domain Controllers - Isolation of perception, planning, and control stacks with predictable timing behavior.

Digital Cockpits & IVI - Safe coexistence of infotainment systems alongside safety-critical vehicle functions.

Vehicle Gateways - Efficient separation between in-vehicle networks and external connectivity interfaces.

Body & Comfort Controllers - Consolidation of traditionally distributed ECUs into zonal architectures.

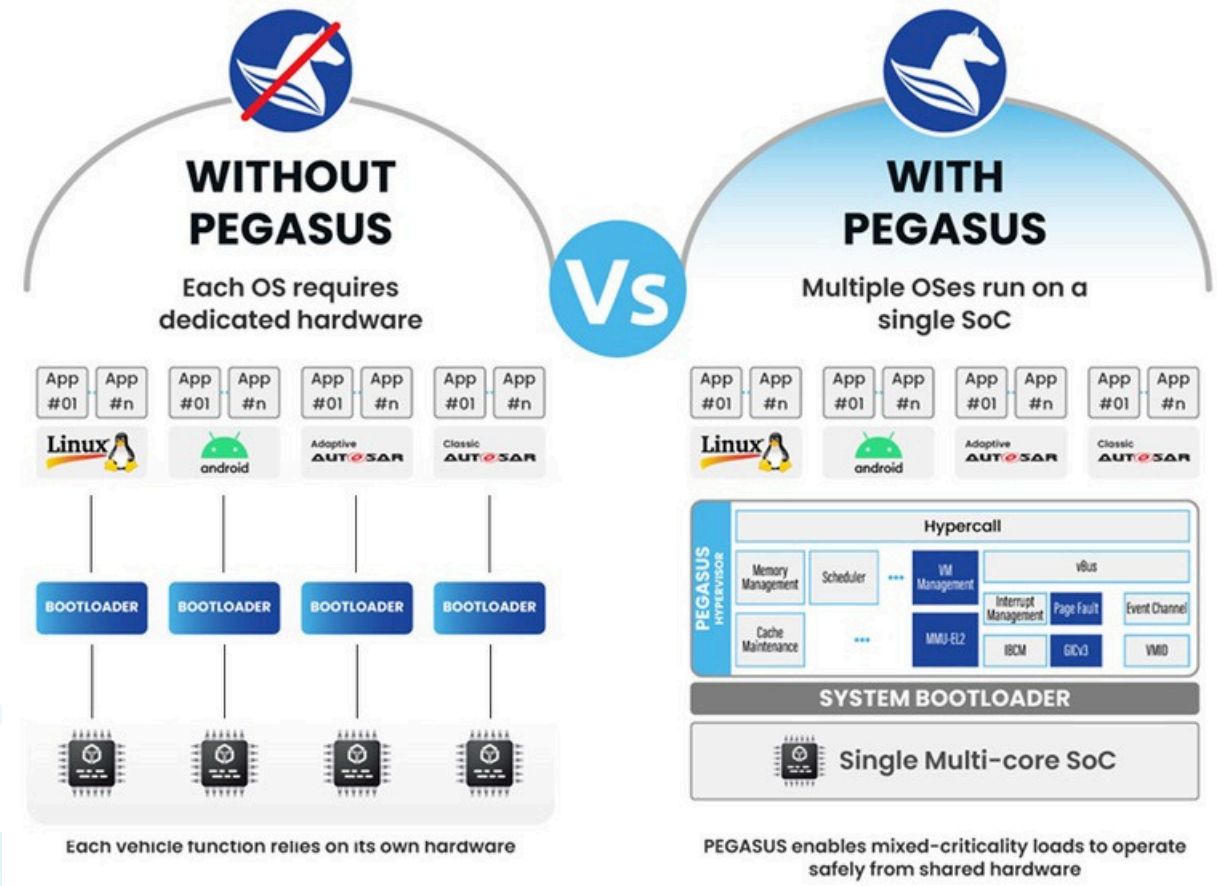
EV Control Systems - Battery management, charging, and energy control domains with strict safety requirements.

Future Mixed-Criticality SDV Platforms - Long-lifecycle platforms supporting incremental software deployment without hardware redesign.

PEGASUS Hypervisor Architecture

PEGASUS is architected as a minimal, safety-oriented hypervisor with clearly defined internal components:

- **VM Manager** – Virtual machine lifecycle management
- **vCPU Manager & Scheduler** – Mapping and scheduling of virtual CPUs onto physical cores
- **Memory Manager** – Static memory partitioning and protection enforcement
- **Hypercall Interface** – Controlled guest-to-hypervisor interaction
- **Virtual Bus & VirtIO Layer** – Efficient virtualized device access
- **Fault Handler Dispatcher** – Fault detection, isolation, and routing
- **Instruction Emulator** – Handling of exceptional or trapped instructions
- **Logger** – System-level logging and observability



Hardware abstraction encompasses CPU cores, memory protection units, access protection units, interrupt controllers, timers, and serial interfaces, enabling PEGASUS to operate consistently across heterogeneous automotive hardware platforms.

Complimentary Tooling

PEGASUS SDK | Workbench & User Dashboard

Workbench (part of the PEGASUS SDK) is the VM configuration and system build environment for PEGASUS-based platforms to target HW. It is designed to reduce the complexity of building, validating, and maintaining virtualization-based automotive systems.

User Dashboard (part of the PEGASUS SDK) provides runtime visibility and operational control at the hypervisor layer. It enables engineers to monitor system status, observe resource usage, and help manage virtual machine lifecycles in real time.

Used together, Workbench and User Dashboard support development, testing, and operational phases of PEGASUS-based platforms, allowing SDV engineering teams to focus on system design and performance rather than tooling complexity

Workbench (part of the PEGASUS SDK)

Workbench is the VM configuration and system build environment for PEGASUS- based virtualization platforms. It is designed to reduce the complexity of building, validating, and maintaining virtualization- based automotive systems throughout their lifecycle.

Workbench enables engineers to work at the system architecture level, focusing on how virtual machines, operating systems, and hardware resources are composed, rather than on low-level configuration mechanics. Through a GUI-driven workflow and automated generation process, Workbench transforms system design intent into consistent, deployable hypervisor images suitable for production SDV platforms.

Workbench interfaces directly with the PEGASUS hypervisor layer, ensuring that system configurations are applied consistently and enforced at the lowest level of the virtualization stack.

Design-Time System Definition and Validation

Workbench supports design-time definition and validation of virtualization configurations, allowing issues to be identified and resolved before deployment.

Using Workbench, engineers can:

- Define overall system topology and VM layouts
- Allocate CPU cores, memory regions, and devices to each VM
- Configure isolation boundaries and access permissions
- Validate configurations to detect conflicts or missing assignments

By enforcing structured configuration models and validation rules, Workbench reduces the likelihood of runtime errors and simplifies safety analysis, integration, and long-term maintenance.

Need to discuss your SDV architecture needs?
sales@cyberperseus.com

Automated Image Generation and Repeatable Builds

Once a system configuration has been defined and validated, Workbench generates consistent, deployable hypervisor system images through an automated build process.

Key characteristics include:

- Automatic generation of manifest files describing VM and resource configuration
- Compilation of validated configurations into binary formats recognized by the hypervisor
- Production of hypervisor binaries ready for deployment on target hardware

This automation supports repeatable system builds, reducing configuration drift between development, testing, and production environments.

Integration into Engineering Workflows

Workbench is designed to integrate into existing engineering toolchains and workflows.

In addition to its GUI-based configuration environment, all Workbench functionality is accessible through a command-line interface (CLI). This enables:

- Integration with build automation systems
- Use within CI/CD pipelines
- Consistent system builds across teams and projects

By standardizing system configuration and build processes, Workbench reduces engineering overhead and minimizes human error in complex SDV programs.

User Dashboard (part of the PEGASUS SDK)

Runtime Visibility and Operational Control for PEGASUS-Based Systems.

User Dashboard provides runtime visibility and controlled operational interaction at the hypervisor layer. It is designed to support engineers during integration, validation, testing, and system operation, offering direct insight into how PEGASUS-based platforms behave once deployed and running.

Where Workbench defines and validates system architecture at design time, the User Dashboard exposes how that architecture behaves in real execution, under load, fault conditions, and evolving system states.

This separation reinforces PEGASUS' design-time vs runtime responsibility model, ensuring that architectural decisions remain stable while operational behavior is observable and manageable.

Core Capabilities

Real-time monitoring of system and virtual machine state - Engineers can observe VM lifecycle status, execution state, and health directly at the hypervisor level, without relying on guest OS instrumentation or application-level logging.

Live visualization of resource utilization - CPU core usage, memory consumption, and resource allocation are visible in real time, allowing engineers to understand how workloads interact under actual operating conditions.

Controlled VM lifecycle operations - The dashboard supports defined runtime actions - such as starting, stopping, resetting, adding, or removing virtual machines - within the boundaries established at design time. These controls support testing, diagnostics, and operational management without altering system architecture.

Observation of system behavior under load and stress - By exposing hypervisor-level metrics and state transitions, User Dashboard allows engineers to validate isolation guarantees, resource partitioning, and performance behavior during peak load, fault injection, or mixed-criticality scenarios.

Role in the PEGASUS Toolchain

User Dashboard is deliberately not a configuration or design tool. It does not redefine system topology, resource allocation, or safety boundaries. Instead, it provides transparent observability and controlled interaction with a system whose structure has already been validated and locked in via Workbench (part of the PEGASUS SDK).

Used together:

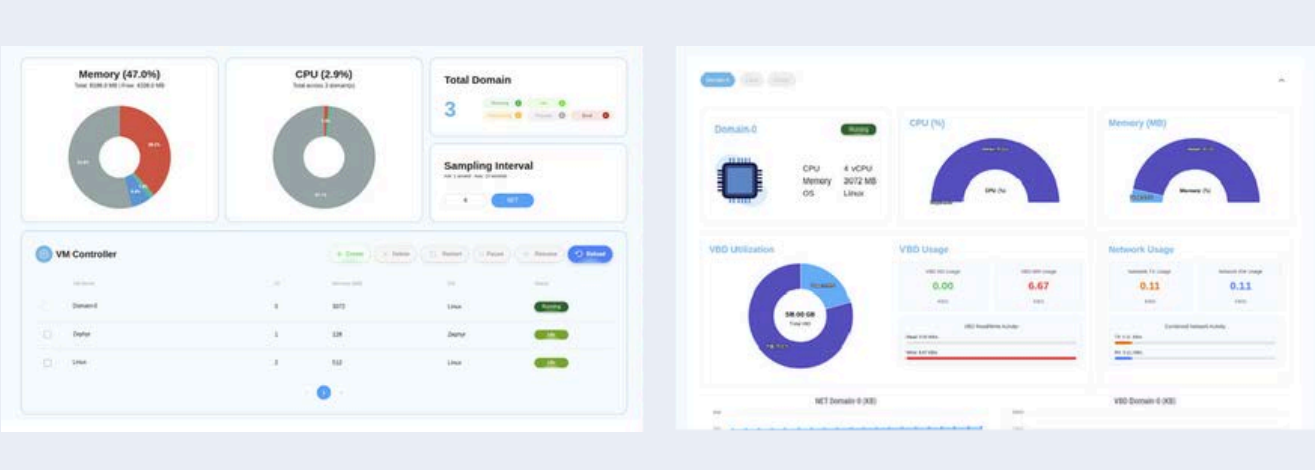
- **Workbench** establishes what the system is allowed to be
- **User Dashboard** shows how the system actually behaves

This pairing allows SDV engineering teams to focus on system behavior, performance, and correctness, rather than building ad-hoc tooling or relying on fragmented, OS-specific monitoring approaches.

Value for SDV Engineering Teams

By operating at the hypervisor layer, the User Dashboard provides a single, authoritative view of system behavior across heterogeneous software domains. This reduces reliance on guest-specific tools, simplifies cross-domain debugging, and improves confidence in system-level properties such as isolation, determinism, and resource control.

In production-oriented SDV programs - where platforms must remain observable, diagnosable, and maintainable over long lifecycles - User Dashboard plays a critical role in ensuring that runtime behavior remains understandable and predictable, without compromising the architectural integrity of PEGASUS-based systems.



AEGIS
SECURE HYPERVISOR SOLUTION

Addressing Security Challenges, AEGIS protects SDV E/E systems from cyber-attack, including DDoS.

The novel access control theory upon which AEGIS' Secure Hypervisor is based, was originally developed by Perseus founder and published in 2009 at a world-class conference.
It has been patented for system seurity of mobility systems.

As vehicles transition to Software-Defined Vehicle (SDV) architectures, increasing connectivity, OTA updates, and workload consolidation **significantly expand the attack surface** of in-vehicle systems. Functions with different safety, security, and real-time requirements—such as infotainment, connectivity, ADAS, and control—are increasingly deployed on shared hardware platforms.

Traditional OS-level security mechanisms and TrustZone-only approaches are insufficient to address these challenges in consolidated SDV systems. **Security must be enforced below the operating system layer**, where resource allocation, isolation, and execution control are determined.

AEGIS addresses these challenges as **foundational system software**, strengthening security at the hypervisor layer while preserving system performance and deterministic behavior.

Introducing AEGIS

AEGIS is Perseus' secure hypervisor solution, designed to extend and harden PEGASUS-based virtualization environments. Rather than acting as a general-purpose hypervisor, AEGIS introduces security-focused mechanisms that protect safety-critical workloads from cyber threats originating in non-critical domains.

AEGIS enables the safe coexistence of multiple operating systems and applications on shared hardware while enforcing strong isolation, controlled resource usage, and privileged access boundaries. It is designed for SDV platforms where efficiency, security, and performance must be balanced at the system architecture level.

Core Security Capabilities

Strict Isolation and Secure Partitioning - Safety-critical OS domains are isolated from non-critical OS domains such as infotainment and connectivity, preventing from fault or compromise propagation caused by threats such as SW bugs or denial-of-service (DoS) attacks.

Fine-grained Resource Control – CPU/IO scheduling and resource allocation are enforced at the hypervisor layer to ensure that critical functions maintain deterministic performance, including under adverse conditions such as DoS attacks.

Privilege Access Control - Inter-VM communication is explicitly controlled, reducing the risk of unauthorized access, lateral movement, and data leakage.

System-Level Threat Containment - Security enforcement occurs below the OS, allowing compromised applications or operating systems to be contained without impacting critical vehicle functions.

Applications in Automotive Systems

Cybersecurity for Connected Vehicles - Protects in-vehicle networks and inter-application communication from external and internal threats such as DoS attacks or SW bugs.

ADAS and Safety-Critical Control Systems - Preserves deterministic behavior and isolation for functions with strict safety and timing requirements.

Infotainment and Telematics - Enables Linux- and Android-based systems to coexist with safety-critical software without compromising system integrity.

OTA-Enabled SDV Platforms - Supports secure lifecycle operation by isolating updateable software components from critical execution domains.

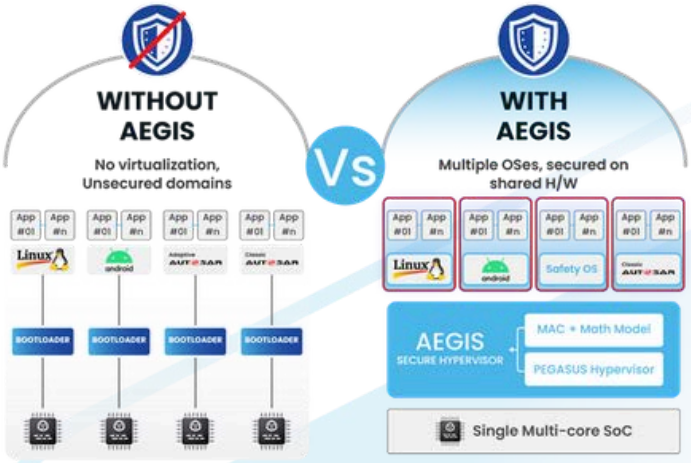
Secure Vehicle Gateways - Secure separation between in-vehicle networks and external connectivity interfaces.

Role Within the Perseus Software Suite

AEGIS complements PEGASUS by strengthening security at the virtualization layer and works alongside AEGIS-let, GAIA, and TACHYON to form a cohesive foundational system software stack for SDV platforms.

Within this architecture, AEGIS specifically addresses safety, security and cyber resilience, ensuring that SDV systems can scale in complexity without sacrificing safety, performance, or long-term maintainability.

AEGIS Architecture



TACHYON
LINUX FAST BOOT SOFTWARE

Boot Linux For Automotive Systems in Under 1.5 Seconds

Addressing Slow Boot Times in Automotive Linux Systems

Modern vehicles increasingly rely on **Linux-based operating systems** to support both safety-critical and non-critical functions, including **rearview cameras, instrument clusters, infotainment systems, head-up displays (HUDs), ADAS and telematics platforms**.

While Linux provides a rich ecosystem and long-term flexibility, conventional automotive Linux implementations suffer from slow system startup, creating safety, compliance, and usability challenges.

In traditional deployments, Linux-based automotive systems can require up to 15 seconds or more to become fully operational after ignition. As vehicles move toward Software-Defined Vehicle (SDV) architectures, such delays are no longer acceptable.

Safety Risks Associated with Delayed Boot-Up

Certain vehicle functions must be available immediately at startup to ensure safe operation. Rearview cameras, for example, are required during reversing and parking maneuvers, where delayed activation directly increases the risk of accidents.

Function	Use Case	Safety Impact
Rearview camera	Reversing, parking	Increased collision risk
Instrument cluster	Driver awareness	Reduced situational awareness
HUD	Speed, alerts	Delayed warnings
ADAS visual feedback	Low-speed assist	Partial system availability

Regulatory and Compliance Requirements

Global safety regulations impose strict requirements on startup latency for safetyrelated vehicle systems:
UNECE Regulation No. 158 (EU) mandates that rearview camera systems must become operational **within two seconds of vehicle ignition.**
NHTSA regulations (USA) enforce comparable activation time requirements for backup camera systems.
Failure to meet these requirements can result in **regulatory non-compliance, recalls, financial penalties, and legal exposure for OEMs.**

User Experience and Market Expectations

Beyond regulatory compliance, driver expectations have evolved. Consumers increasingly expect instant system responsiveness, comparable to smartphones and modern consumer electronics. Delayed startup of displays, cameras, or infotainment systems negatively impacts perceived vehicle quality, brand reputation, and customer satisfaction.

RTOS - A Partial Solution, With Limitations

Real-Time Operating Systems (RTOS) can achieve near-instant startup due to their minimal design and deterministic execution. However, RTOS-based approaches introduce significant trade-offs for SDV programs:

Specialized Engineering Requirements - RTOS development demands expertise in real-time scheduling, embedded safety, and low-level system integration.

Limited Ecosystem Support - Unlike Linux, RTOS platforms typically rely on proprietary toolchains, middleware, and vendor-specific frameworks, which indicates lack of scalability in applications like GUI.

Integration Complexity - Integrating RTOS-based components with Linux-based ADAS, infotainment, and connectivity stacks often requires custom drivers and interfaces.

Increased Cost and Development Time - RTOS-centric architectures increase engineering effort, integration timelines, and long-term maintenance costs.

Introducing TACHYON - Fast Boot Performance Without RTOS

TACHYON is Perseus' Linux Fast Boot solution, developed as foundational system software for SDV platforms that require RTOS-like startup performance while remaining within the Linux ecosystem.

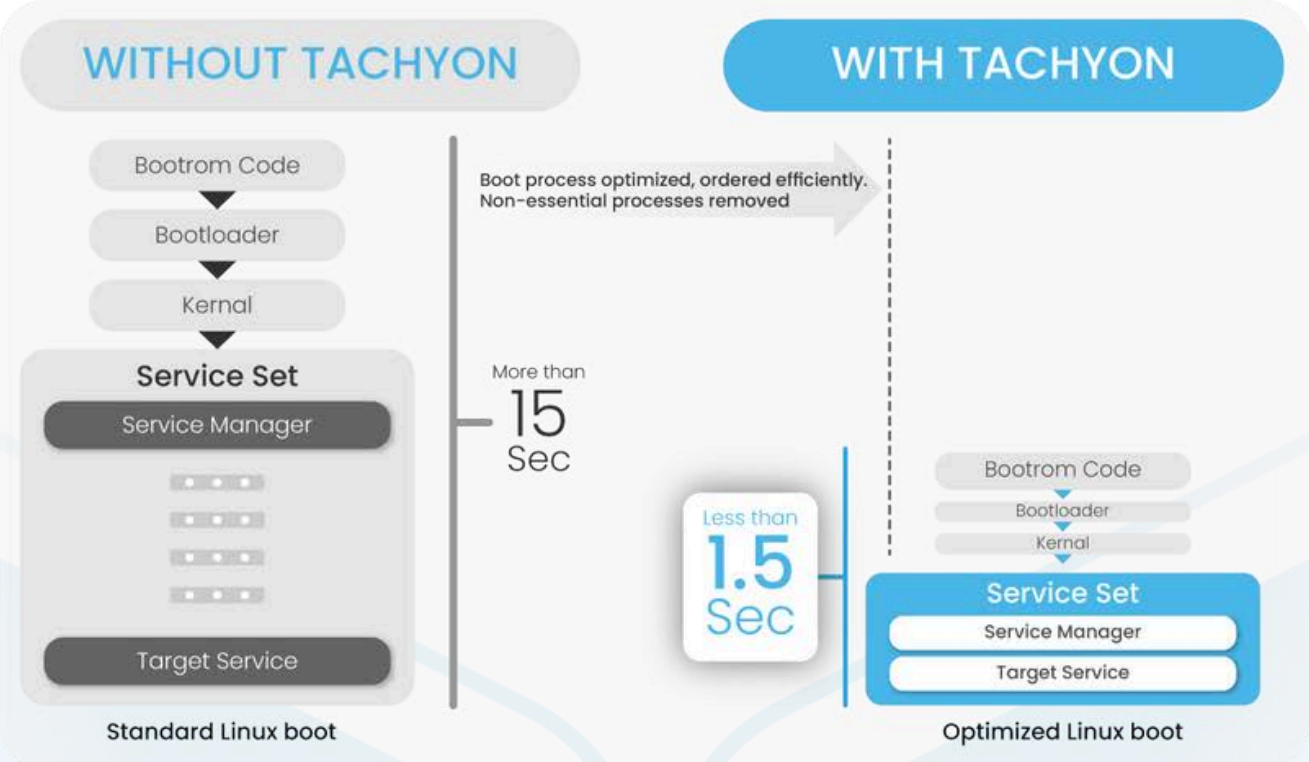
For example, in the case of rearview camera apps: By eliminating non-essential boot stages and optimizing system initialization, TACHYON enables Linux-based automotive systems to reach operational readiness in **under 1.5 seconds**, supporting both regulatory compliance and improved user experience.

Core Capabilities

The Ultra-Fast Linux Boot significantly **reduces the Linux kernel startup time to as low as 0.7 seconds** while **meeting UNECE R158 and NHTSA activation requirements**. This innovation allows for a rapid startup without altering applications or discontinuing the use of Linux.

The system features optimized initialization through a lightweight bootloader and kernel-level enhancements, **prioritizing the startup of safety-critical components** such as rearview cameras and clusters. Non-essential services are deferred until these critical functions are operational.

Designed specifically for **ARM Cortex-A–based automotive SoCs**, the integration supports standard automotive Linux distributions. It eliminates the need for RTOS migration or the adoption of proprietary boot frameworks.

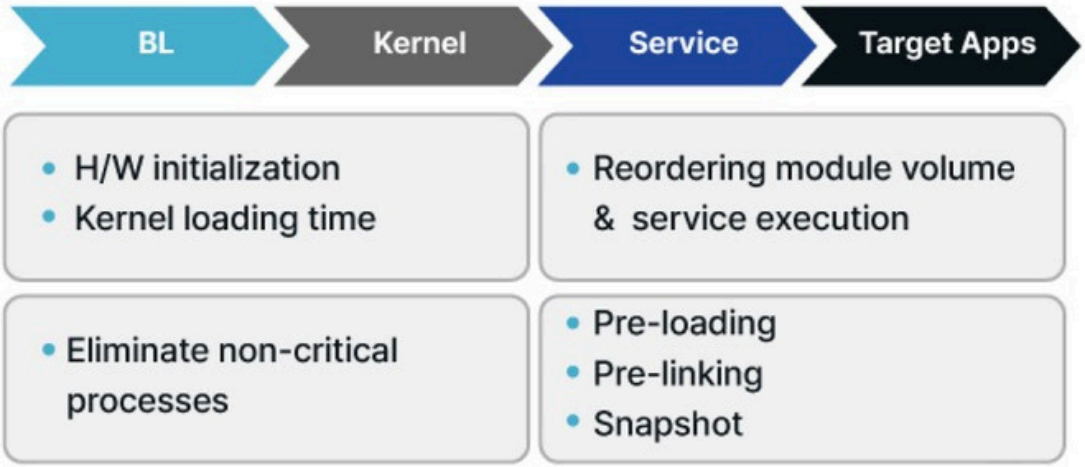


Applicability Across Automotive Systems

Although originally developed to address rearview camera latency, TACHYON applies broadly across Linux-based vehicle systems, including:

- **Linux Infotainment systems** — faster access to navigation, media, and connectivity
- **Android Infotainment systems** — faster access to navigation, media, and connectivity
- **Head-up displays (HUDs)** — immediate availability of driver information
- **Telematics and connectivity platforms** — reduced delay in diagnostics, data exchange, and cloud services
- **ADAS systems** — reduced delay in sensor data acquisition, diagnostics, and decision-making services

How it works



Cost and Development Implications

TACHYON delivers performance improvements **without restructuring existing software stacks**. Unlike RTOS-based approaches, it allows OEMs and suppliers to:

- Avoid specialized RTOS engineering costs
- Shorten development and deployment timelines
- Reuse existing Linux software assets
- Reduce integration and validation effort

Role in Software-Defined Vehicle Architectures

As SDV platforms evolve toward **centralized and zonal computing**, the demand for immediate system responsiveness continues to increase. TACHYON addresses this requirement at the system-software level, enabling:

- Compliance with global safety regulations
- Fast availability of safety-critical functions
- Improved driver experience
- Predictable startup behavior across vehicle lifecycles

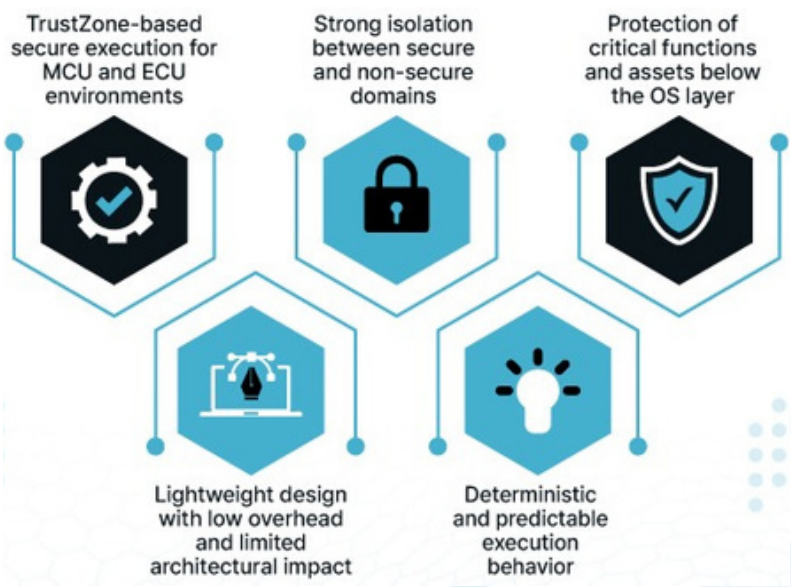
TACHYON operates alongside PEGASUS and GAIA as part of Perseus' foundational system software suite, addressing performance and efficiency at system startup without introducing additional architectural complexity.

AEGIS-let

Embedded Secure Kernel for Automotive Security

As connected and autonomous vehicles become increasingly software-driven, ensuring secure operation of electronic control units (ECUs) is critical. Automotive systems face evolving cyber threats, requiring robust security measures to isolate sensitive data and protect critical functions from malicious attacks.

AEGIS-let, is a lightweight, high-performance secure kernel designed specifically for ARM TrustZone environments on Cortex-A, Cortex-M, and Cortex-R52 cores. It provides a hardware- enforced secure execution environment, ensuring data integrity, process isolation, and secure resource management for modern automotive architectures.



AEGIS-let is a lightweight, highly efficient secure kernel designed to protect automotive ECUs from cyber threats, ensuring safe execution of critical software.

By leveraging ARM TrustZone, enforcing strict process isolation, and providing secure communication frameworks, AEGIS-let enables automakers and Tier 1 suppliers to build secure, scalable, and futureproof vehicle architectures.

Key Security Challenges in Automotive Systems

Increasing Cyber Threats in Connected Vehicles – As vehicles become more connected, they are vulnerable to remote exploits, unauthorized data access, and system manipulation.

Need for Secure Process Isolation – Automotive ECUs run multiple applications, requiring strict isolation between secure and non-secure operations to prevent unauthorized access.

Efficient Resource Management – Security solutions must be lightweight and efficient to run on memory-constrained automotive hardware without affecting performance.

How AEGIS-let Addresses These Challanges

Hardware-Based Security Enforcement – Leverages ARM TrustZone technology to create a secure execution area, isolating critical processes and data.

Lightweight Secure Kernel – Provides essential security functions without excessive resource consumption, making it ideal for real-time automotive applications.

Process and Privilege Isolation – Ensures that secure applications run separately from general-purpose automotive software, preventing unauthorized access.

Secure Communication & Exception Handling – Manages internal ECU communication, I/O operations, and security protocols within a trusted execution environment.

Dynamic Secure Application Execution – Supports external ELF file-based application loading, enabling flexibility in deploying security-critical functions.

Key Features

Multi-Core Support – Operates on Cortex-A, Cortex-M, and Cortex-R52 cores, supporting a wide range of automotive ECUs.

Secure Device Driver Framework – Ensures that only authorized firmware and drivers are executed within the secure environment. Sealed Storage (Optional depending on HW availability) – Protects sensitive automotive data from tampering or unauthorized access.

Privilege Isolation – Enforces separation between kernel and application privileges, reducing the risk of unauthorized system control.

General-Purpose TEE API (In Development) – Future support for Trusted Execution Environment (TEE) APIs, allowing integration with standardized secure automotive software frameworks.

Multi-Language Support – Developed in C++ and Rust, ensuring high performance, security, and flexibility for automotive applications.

Real-World Applications

AEGIS-let serves as the security foundation for multiple automotive use cases, ensuring safe, reliable, and tamper-proof operation of vehicle systems:

- **Powertrain & Safety-Critical Systems** – Secure execution of engine control, braking, and airbag ECU software, ensuring protection against cyber threats.
- **Autonomous Driving Systems** – Isolates AI-driven decision-making processes to prevent unauthorized interference.
- **Secure Communication Gateways** – Protects vehicle-to-vehicle (V2V) and vehicle-to-cloud (V2C) communication from data breaches.
- **In-Vehicle Payment & Digital Keys** – Ensures safe authentication and encryption for contactless payment systems and digital key solutions.
- **Over-the-Air (OTA) Security** – Prevents unauthorized firmware modifications during software updates.

Technical Specifications

Feature	Specification
Binary Size	< 128KB (Kernel Only)
Supported Architectures	ARM V8/V9 TrustZone of Cortex-A, Cortex-R or Cortex-M (Cortex-M33, M7), etc
Memory Requirement	DRAM or SRAM (8KB)
Dynamic Application Launching	External ELF File
Application Isolation	Yes
Privilege Isolation (Kernel vs. App)	Yes
Secure Device Driver Support	Yes
General-Purpose TEE API	In Development
Sealed Storage	Optional depending on HW availability
Programming Languages	Rust, C++

GAIA

Universal & Advanced Secure Boot Loader for Automotive Systems

Tier 1 and Tier 2 automotive suppliers face major challenges when integrating third-party systems into OEM architectures, which often lack standardization across brands—and even within different models of the same brand. This inconsistency leads to inefficiencies, higher development costs, and compatibility issues in software and hardware integration.

GAIA addresses these challenges by providing a versatile, 'one-size-fits-all' boot loader solution, ensuring seamless communication between the hardware layer (MCU) and the operating system (OS) layer. It enables efficient data transfer, optimizes boot performance, and enhances security—making it a critical component in modern automotive architectures.

Integration Challenge	The industry faces higher cost and complexity due to inconsistent OEM system architectures.
GAIA's Role	GAIA enables reliable MCU-to-OS communication with optimized boot and security.
System Impact	GAIA reduces boot conflicts and improves security and multi-OS scalability.

GAIA is a game-changing boot loader solution that addresses critical challenges in automotive system integration, secure booting, and multi-OS compatibility. By streamlining hardware- software communication, eliminating boot conflicts, and enhancing system security, GAIA empowers automakers and suppliers to develop next-generation vehicles with greater efficiency, security, and scalability.

Understanding MCU Secure Boot Loaders in Modern Vehicles

An MCU boot loader is a specialized program that initializes hardware components and loads the main application software during startup. In connected and autonomous vehicles, its role is critical in ensuring:

Software Integrity Verification – Ensuring that all critical system components are properly initialized.

Cybersecurity & Secure Execution – Protecting the vehicle's software environment by authenticating and verifying firmware and OS components.

OTA (Over-the-Air) Update Support – Enabling secure remote firmware updates while preventing unauthorized modifications.

Universal Boot Loader for Vehicle System Integration

Key Features & Benefits:

- **Seamless Hardware** - Software Integration – GAIA acts as a bridge between the OS and hardware layer, ensuring consistent data flow, stability, and smooth operation of in-car systems.
- **Cross-Platform Compatibility** – Works across multiple OEM architectures, reducing integration time and effort for Tier 1 and Tier 2 suppliers.
- **Hypervisor Compatibility** – Fully integrates with Perseus' Hypervisor solutions, further optimizing system performance and security.
- **Enhanced System Reliability** – Manages data transfer efficiently to ensure critical automotive functions operate without failure.

By simplifying integration and ensuring smooth data communication across platforms, GAIA enables suppliers to streamline development, lower costs, and improve the reliability of automotive systems.

Key Challenges for Automotive Boot Loaders

Lack of Standard Boot Loaders from MCU Vendors – Unlike traditional OS environments, MCU chipset vendors do not provide boot loaders, making integration difficult. The only alternative has been a very few third-party providers, which lacks flexibility in delivery timing for complex automotive systems.

Boot Loader Conflicts in Multi-OS Environments – Modern automotive architectures require multiple OS instances, but legacy built-in OS boot loaders are not designed to coexist. Without a proper solution, legacy boot loaders can collide, causing hardware initialization issues.

Fragmented Software Ecosystems – Automakers often use a mix of AUTOSAR Classic, RTOS, and Hypervisors, requiring a unified boot loader that supports multiple operating systems on a single MCU.

How GAIA Solves These Challenges

Universal Boot Loader

GAIA enables multiple OS environments to run from a single SoC, preventing boot conflicts and optimizing hardware initialization.

MCU-Agnostic Solution

Unlike proprietary solutions, GAIA is not tied to a specific vendor, making it the only independent, flexible alternative for Tier 1 and Tier 2 suppliers.

Integration with Perseus Hypervisors

When paired with Perseus' Hypervisor, GAIA allows multiple virtual machines (VMs) to operate on a single MCU, reducing hardware dependencies and improving system efficiency.

A one-Stop-Shop Solution

By combining hypervisor technology, OS, and secure boot solutions, GAIA ensures a fully integrated approach to SDV development.

KEY FEATURES

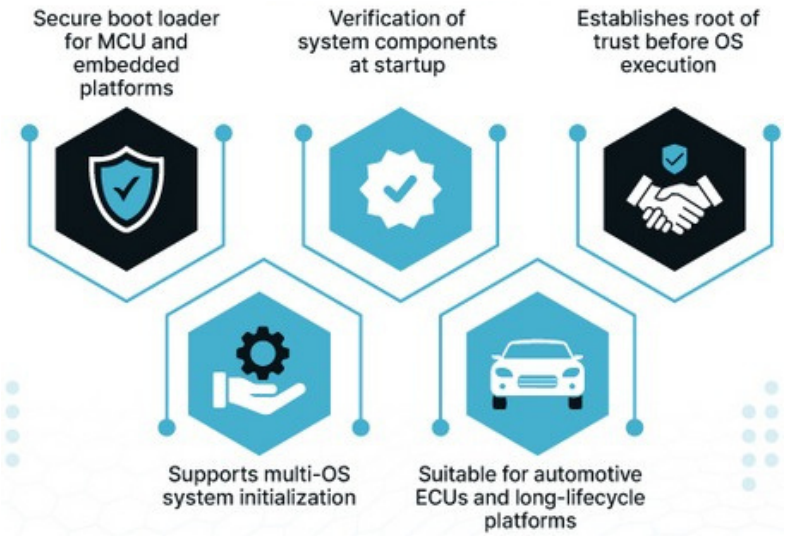
Multi-OS & Hypervisor Support – Supports AUTOSAR Classic (OSEK), RTOS, and Hypervisors, allowing flexible software configurations.

Secure Bootloader Development – Ensures that only authenticated and verified firmware / OS / hypervisor components are executed, enhancing system security.

Hardware Manual & Reference Code – Includes detailed documentation and reference implementations, streamlining the development and testing of evaluation boards for new automotive architectures.

GAIA System Architecture

GAIA's architecture combines boot loader and hypervisor layers on a multicore automotive MCU, enabling secure boot, hardware initialization, multi-OS isolation, efficient interrupt handling, and predictable startup for SDV platforms.



Application of GAIA in SDV architectures

Real-World Applications

GAIA is designed for a wide range of automotive applications, including both next-generation connected vehicles and traditional ECUs such as:

ADAS & Autonomous Driving Systems – Ensures fast and secure initialization of critical perception, decision-making, and control functions.

Chassis, Body, & Powertrain ECUs – Guarantees that essential firmware and OS boot processes execute correctly, improving vehicle reliability.

EV Battery Management Systems (BMS) – Supports real-time, high performance boot loading for battery monitoring and optimization.

Technical Specifications

Feature	Specification
Supported Architectures	ARM V8/V9 Cortex-R52 MCU, Infineon AURIX TC4x MCU, STMicro R-52 Multicore Stellar, NXP R-52 Multicore S32E/Z, etc.
Binary Size	< 128KB
Memory Requirement	DRAM or SRAM (4KB Stack). No heap memory required.
Secure Boot Support	Yes
Command Line Interface (CLI)	Yes
Binary Download Capability	Yes
Programming Languages	C/C++, Rust

Perseus' Market Impact and Industry Recognition

Perseus works with automotive OEMs and suppliers developing Software-Defined Vehicle (SDV) platforms, providing production-ready foundational system software that operates at the lowest layers of the vehicle software stack—where efficiency, security, and performance are determined.

Perseus solutions are designed, certified, and validated for real vehicle programs, rather than experimental or research deployments. This production readiness reduces platform risk for OEMs and Tier suppliers, shortens integration and validation cycles, and enables SDV architectures to move from concept to series production with predictable safety, performance, and lifecycle behavior.

The company's industry relevance is reflected in a series of verifiable milestones across functional safety certification, production deployment, semiconductor ecosystem integration, and independent validation. Together, these demonstrate Perseus' ability to translate system- level software innovation into deployable, long-lifecycle automotive platforms.

Are your partners ready for the Software-Defined Vehicle era?

We're ready, and we look forward to building the future together.

sales@cyberperseus.com

Selected Industry Milestones

Independent Validation with Fraunhofer IESE (2024)

Perseus partnered with Fraunhofer IESE to co-develop a virtual validation and verification environment for embedded systems.
The collaboration supports system-level testing and certification of complex, virtualized automotive platforms, with the potential to significantly reduce validation time and cost.

ISO 26262 ASIL-D Certification (2025)

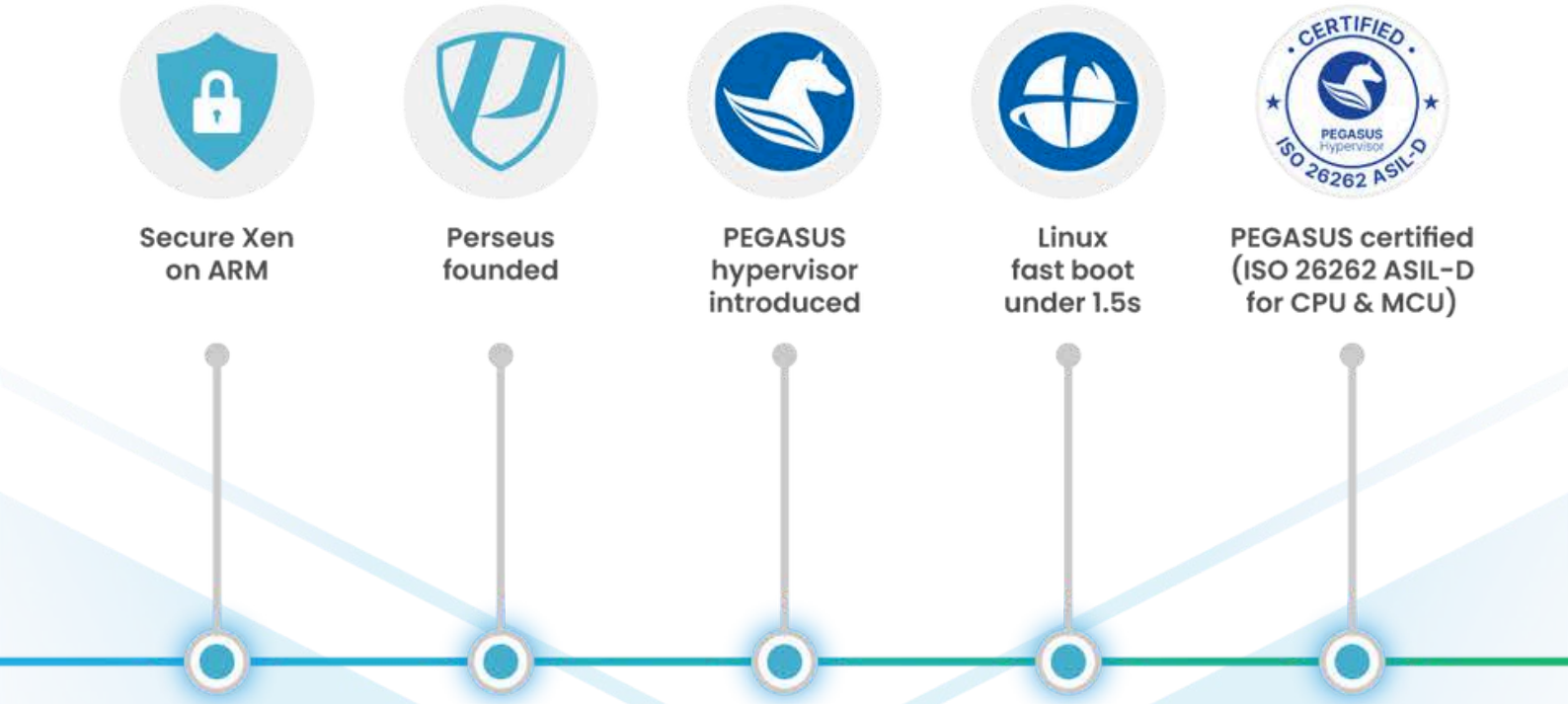
The PEGASUS hypervisor achieved ISO 26262 ASIL-D certification for both CPU and MCU architectures. This validates its suitability for safety-critical SDV platforms and places PEGASUS among a small number of hypervisors certified at the highest automotive functional safety level across both architectural domains.

Semiconductor Ecosystem Integration (2025)

PEGASUS has been integrated with leading automotive semiconductor platforms, enabling pre-validated hypervisor configurations and reducing system integration effort for SDV programs. This supports faster platform bring-up and more predictable system design.

Production Deployment of Linux Fast Boot (2026)

Perseus' TACHYON Linux Fast Boot technology is scheduled for deployment in a production vehicle program, supporting time-critical Linux-based vehicle functions. This milestone demonstrates real-world adoption of fast-boot Linux in safety and compliance-constrained automotive environments.



Perseus R&D Timeline & Company Milestones

Perseus' milestones reflect nearly two decades of deep hypervisor and automotive software expertise, from early Xen ARM innovation to ASIL-D–certified SDV platforms.

The timeline highlights continuous R&D leadership, industry partnerships, and the evolution of secure, high-performance virtualization and fast-boot technologies for next-generation vehicles.

Year	Milestones
2007	Secure Xen ARM architecture introduced at the Xen Summit (North America).
2008	Xen ARM 1.0 released (Xen Hypervisor, Mini OS).Xen ARM 2.0 released with paravirtualized Linux Kernel (v2.6.24) and Xen tools.
2009	Xen ARM 3.0 released with ARM11MP core support.Secure Xen ARM Hypervisor security paper presented at a world-class conference.
2010	Xen ARM 4.0 released with performance optimizations.
2016 (Dec)	Perseus founded, uniting experts from research institutes, open-source communities, and major electronics brands.
2017	Strategic investment secured from Kakao.Launched PEGASUS and AEGIS solutions; multi-OS automotive SoC demo at IAA Frankfurt Motor Show.
2018	Appointed lead for Hypervisor API standardization in GENIVI Alliance (now COVESA).
2019	Launched TACHYON Linux Fast Boot (<1.5s).Invited to the Global Entrepreneurship Summit (U.S. Department of State).
2020	Selected to join Samsung C-Lab Outside program.
2021	Showcased Hypervisor, Secure Container, and TACHYON at CES 2021 (USA) and Automotive World 2021 (Japan).
2023	Selected for Open Innovation Programs by Hyundai ZeroOne and LG Innotek.
2024	Won Mobility Award at TechSpark 2024 (APAC).Secured $8M VC funding. Appointed partner to NXP (Gold), Renesas (Preferred), ARM, SOAFEE, and Infineon Technologies. Joint project with Fraunhofer IESE.
2025	Achieved ISO 26262 ASIL-D certification for PEGASUS Hypervisor (CPU & MCU). Launched PEGASUS Workbench and User Dashboard. Selected for Mercedes-Benz Autobahn Open Innovation Program.

Software-Defined Everything
Future Opportunities Beyond Automotive

The transition toward Software-Defined Vehicles reflects a broader shift seen throughout computing history—from dedicated hardware to virtualization, from on-premise systems to cloud platforms, and from fixed terminals to mobile computing. In each case, control moved from rigid hardware to software-defined layers, enabling greater scalability, flexibility, and faster innovation.

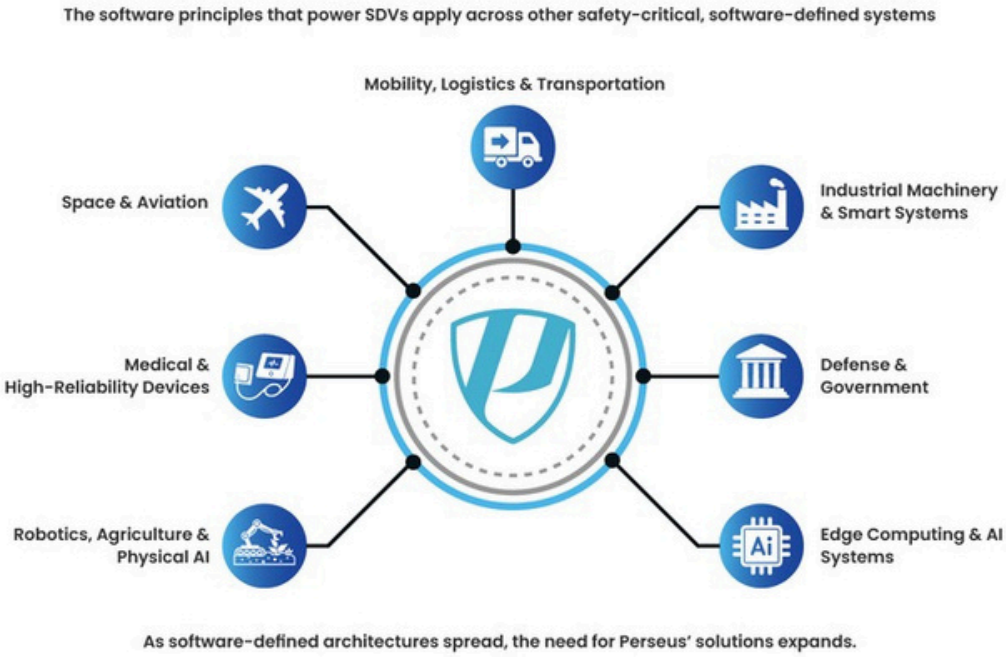
A similar transformation is now underway across many safety-critical industries. Physical infrastructure, heavy machinery, and tightly coupled control systems are increasingly being replaced or augmented by software-defined architectures. Automation abstracts hardware— and often direct human involvement —placing responsibility for coordination, safety enforcement, and real-time decision-making at the system software layer to improve efficiency, security, and performance.

Perseus' approach—delivering foundational system software that enforces isolation, determinism, and efficient resource sharing—directly applies to these environments. As functionality consolidates onto shared compute platforms, technologies such as hypervisors and fast-boot system software become essential infrastructure, enabling scalable, safety- oriented systems well beyond automotive.

Why This Matters

Across these sectors, the same system-level requirements recur: efficient use of shared hardware, strong isolation between workloads, deterministic execution, secure updates, and long-term platform evolution. These are precisely the challenges addressed by Perseus' foundational system software, making its technology and architectural approach relevant far beyond Software-Defined Vehicles.

Applicable Domains Beyond Automotive



Sector	Description
Mobility, Logistics & Transportation	Rail, autonomous shipping, urban air mobility, drones, ports, and airport operations requiring deterministic, isolated, real-time coordination.
Space & Aviation	Aerospace, avionics, satellite, and ground control systems operating under strict safety, reliability, and certification constraints.
Industrial & Critical Infrastructure	Energy, utilities, manufacturing, robotics, heavy equipment, agriculture, mining, and process control requiring determinism and fault containment.
Defense & Government Systems	Defense vehicles, command-and-control, secure communications, surveillance, and mission-critical embedded platforms.
Edge Computing & Physical AI Systems	Edge AI, intelligent gateways, sensor fusion, and distributed control on constrained hardware.
High-Reliability Professional Systems	Medical devices, imaging, laboratory automation, broadcast systems, and scientific instrumentation with long lifecycles.
Smart Cities & IoT Platforms	Traffic systems, building control, public safety platforms, and industrial IoT infrastructure requiring secure distributed control.

Who We Work With

Perseus works across the automotive ecosystem, supporting organizations building and scaling Software-Defined Vehicle (SDV) platforms.

Customers

- Automotive OEMs transitioning from hardware-centric designs to centralized and zonal SDV architectures.
- Tier 1 suppliers developing safety-critical systems such as ADAS, digital cockpits, infotainment, HUDs, and central compute platforms.
- Tier 2 suppliers that provide SoCs (CPU/MCU), embedded software, and platform technologies that must meet strict safety, security, and real-time requirements.

Teams We Typically Engage

SDV and vehicle automation teams (e.g. ADAS, autonomous driving)

Infotainment, cockpit, and connectivity engineering teams

Software safety and cybersecurity teams (e.g. ISO 26262, system isolation)

Embedded systems and hardware platform teams (CPU, MCU, domain/ Zonal controllers)

In many cases, Perseus works closely with supplier organizations that have deep, long-term integration with OEM platforms, where system-level software decisions have the greatest architectural impact.

Reasons to Believe

Global Network and Ecosystem Support

Perseus participates in a broad automotive and embedded-systems ecosystem spanning semiconductor platforms, operating systems, software frameworks, research institutions, and industry programs. The partners and vendors referenced below reflect representative environments in which foundational system software - such as that developed by Perseus - must operate to achieve SDV goals, like long lifecycles, safety-critical requirements, and close alignment with production hardware and operating systems.

As such, the brands listed below are strictly representative, rather than an exhaustive list. Perseus' solutions are are HW and OS diagnostic by design - applicable across the automotive ecosystem, ensuring strong support for diverse OEM, Tier-1, and ecosystem requirements.

Partner of Major Brands

Infineon, NXP Gold Partner, SOAFEE, Fraunhofer IESE, arm, RENESAS, HYUNDAI KEFICO

Hardware Support

Infineon, STMicroelectronics life.augmented, Qualcomm, NXP, NVIDIA, RENESAS, arm

... and more

Software Support

android, Linux, AUTOSAR, QNX, INTEGRITY SECURITY SERVICES, freeRTOS

... and more

Expertise and Accreditation

Perseus' credibility is built on sustained technical leadership, continuous product execution, and demonstrated readiness for production deployment—at the foundational system software layer where efficiency, security, and performance are determined.

Deep, Long-Term Technical Expertise

Perseus' engineering leadership brings more than 18 years of experience in embedded virtualization, real-time systems, and safety-critical software. This background spans open-source leadership, applied research, and production system development, enabling delivery of production-grade foundational software rather than experimental or OS-level solutions.

Consistent Product Execution

Perseus demonstrates ongoing execution through continuous improvement of core technologies and expansion of its system software portfolio. This includes measurable performance gains—such as ongoing reductions in Linux boot time with TACHYON—and the introduction of essential platform tooling, including PEGASUS Workbench (SDK) and the PEGASUS User Dashboard, supporting design, deployment, and operational use at scale.

Standards and Ecosystem Leadership

Since 2018, Perseus has played a leading role in open automotive virtualization initiatives (via COVESA, previously GENIVI), reflecting peer recognition of its architectural and technical leadership.
Active participation across semiconductor and software ecosystems further anchors its solutions in real-world platforms and long lifecycle requirements.

In 2025 Perseus was awarded ISO 26262 ASIL-D certification for its PEGASUS hypervisor, both CPU and MCU architectures.

Certification and Production Readiness

In 2025, the PEGASUS hypervisor achieved ISO 26262 ASIL-D certification for both CPU and MCU architectures, confirming engineering maturity and suitability for safety-critical systems. Perseus solutions are production-ready, supporting OEM and supplier programs transitioning from development to series deployment.

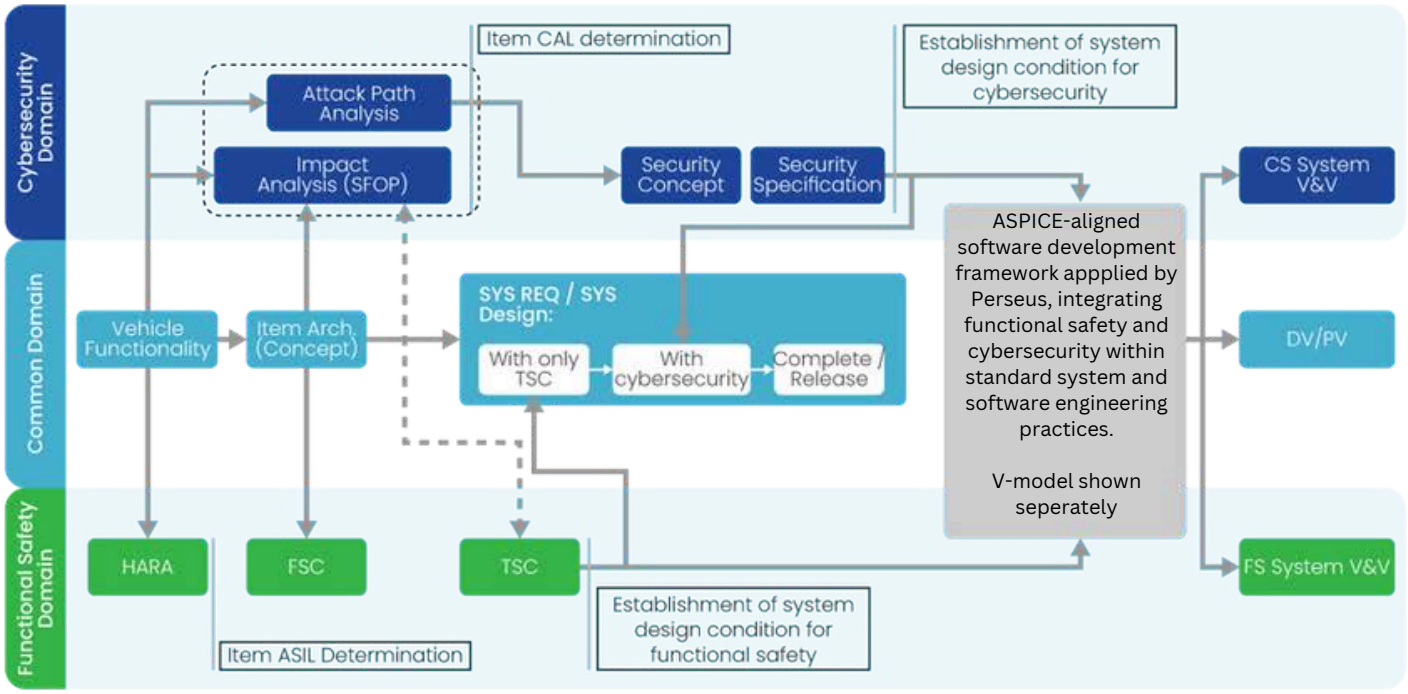
Research Validation and Organizational Continuity

Collaboration with Fraunhofer IESE on virtual verification reinforces Perseus' focus on system-level validation for complex, virtualized platforms. Led by an experienced technical founder with largescale production background, and supported by continued investment through 2024, Perseus is structured for long-term execution across multi-year vehicle programs.

Trusted Software Development Framework and Processes

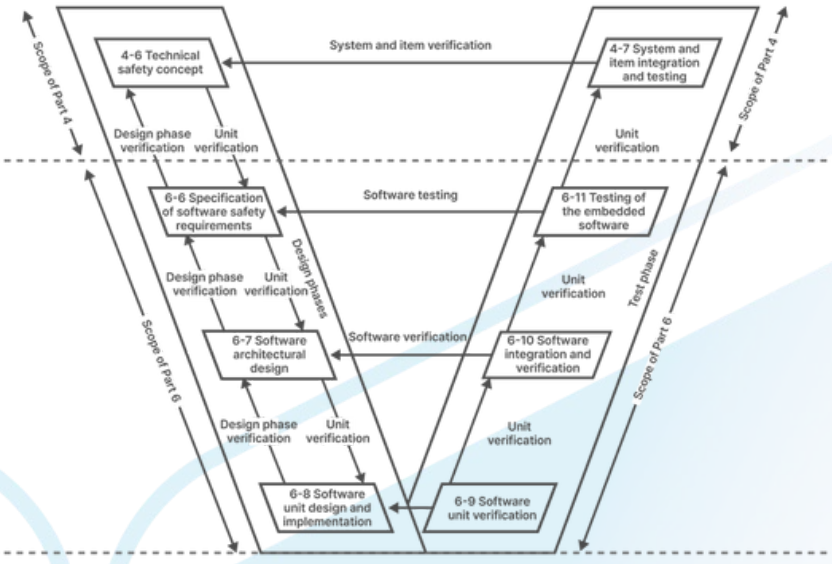
Perseus applies an integrated automotive software development framework across its portfolio, aligning functional safety, cybersecurity, and process maturity into a unified approach that meets OEM and Tier-1 expectations. Rather than treating safety and security as downstream checks, Perseus embeds these disciplines from the earliest system design stages.

Perseus' ASPICE-Aligned Software Development Framework



Functional safety activities—HARA, FSC, and TSC in accordance with ISO 26262—and cybersecurity activities, including TARA, impact analysis, and attack path analysis under ISO/SAE 21434, are performed in parallel. Together, they define explicit safety and security constraints, such as ASIL and CAL determinations, which establish mandatory system design conditions before software development begins. This early constraint definition reduces late-stage risk and supports consistent architectural decisions.

These safety and security constraints converge into a unified system requirements and design baseline, providing a controlled input to software development. Implementation then follows an industry-standard, ASPICE-aligned Vmodel, ensuring end-to-end traceability. Dedicated verification and validation activities confirm standards compliance, audit readiness, and consistent quality from concept through production deployment.



Meet Sang-bum Suh PhD
Founder & CEO, Perseus

Sang-bum Suh is a systems software engineer and founder of Perseus, with more than 30 years of experience in embedded systems, virtualization, security and parallel computing.

He founded Perseus to build foundational system software for Software-Defined Vehicles, addressing efficiency, security, and performance at the system architecture level.

Dr. Suh is a pioneer of embedded virtualization. In 2007, he founded Secure Xen ARM, one of the earliest open-source hypervisor software community projects for ARM-based embedded systems, and led the Xen ARM community for nearly a decade. This work established key architectural principles for the pioneering secure virtualization in resource-constrained and safety-critical environments and laid the groundwork for Perseus' automotive hypervisor and security technologies.

He holds more than 70 U.S. patents spanning secure virtualization, multicore systems, and security technologies, and has published research in venues including ACM MobiCom, PACT, and IEEE CCNC. He has also presented at international technical conferences such as Xen Summit.

Before founding Perseus, Dr. Suh spent 13 years at Samsung Electronics, including serving as Vice President, where he led system software and security architecture for the Tizen SW platform used in Samsung's wearable watches and globally leading smart TVs.
He earned a PhD degree in Computer Science from the University of Cambridge.

Q&A With Sang-bum Suh PhD

What fundamentally differentiates Perseus in automotive system software?

Perseus focuses on foundational system software—the layer below operating systems and applications where efficiency, security, and performance are actually determined. Architectural decisions made at this level are very difficult to change later. With more than 18 years of experience in embedded virtualization and secure systems, we deliver production- ready software with certification, tooling, and long-term lifecycle support, not just technical concepts.

What does “Foundational System Software” mean in the context of Software-Defined Vehicles?

It refers to software that decouples functionality from specific hardware. Hypervisors are a prerequisite for Software-Defined Vehicles because they allow workloads to evolve independently of silicon. Historically, safety-critical and non-critical functions had to run on separate chips, which does not scale. Perseus enables both to run on shared hardware while still meeting strict safety and real-time requirements.

Why is ISO 26262 ASIL-D certification for PEGASUS such a significant milestone?

ASIL-D certification validates both the product and the engineering discipline behind it. It confirms PEGASUS is suitable for safety-critical production systems, not experimental platforms. Certification across both CPU and MCU architectures is especially important because modern SDVs combine application-class and real- time workloads on shared platforms.

How does the Fraunhofer IESE partnership strengthen Perseus’ technology?

Hypervisors are still relatively new in automotive, and not all future use cases are known. Fraunhofer helps us prove that PEGASUS is safe, effective, and future- proof as new applications emerge. Their virtual verification platform allows industry-recognized testing before physical hardware exists, potentially reducing validation timelines from around 18 months to closer to six.

How do you view the future of hypervisors in SDVs and beyond?

We are still early in the SDV transition. Hypervisors are the enabling technology that allows new software capabilities to emerge without redesigning hardware every time. Efficiency comes from decoupling software from hardware, security comes from isolation, and performance comes from deterministic control. Those principles apply beyond automotive, but SDVs are where the constraints are most visible today.

What is Perseus working on today?

Building on automotive foundations, we’re extending core system software into other safety-critical industries where determinism, isolation, and long lifecycle support are required, while maintaining a strong focus on R&D. Target future arenas include industrial, aerospace, defense, and edge-AI platforms. In parallel, Perseus is refining foundational tooling and platform integration to support mixed-criticality workloads and production-scale deployments beyond automotive.

FAQ

How do organizations typically engage with Perseus?

Perseus works directly with automotive OEMs, Tier-1 and Tier-2 suppliers, and ecosystem partners on Software-Defined Vehicle (SDV) platforms. Engagements usually begin with an architectural discussion rather than a product demo, because Perseus' solutions operate at the foundational system software layer and influence long-term platform decisions. Integration is typically incremental, aligned with existing vehicle architectures.

What platforms and operating systems does Perseus support?

Perseus is designed to work within existing automotive stacks. It supports application-class CPUs and real-time MCU platforms, including ARM Cortex-A, ARM Cortex-R, selected RISC-V platforms, and leading automotive SoCs (including Infineon Technologies).
Supported operating systems include AUTOSAR Classic, AUTOSAR Adaptive, RTOSs, Linux, Android, QNX, and legacy systems, enabling mixed-criticality workloads to coexist on shared hardware.

How does Perseus approach security and functional safety?

Security is enforced below the operating system layer through architectural isolation, privilege control, and deterministic resource management. This system-level approach limits fault propagation and contains cyber threats.
PEGASUS, Perseus' automotive hypervisor, achieved ISO 26262 ASIL-D certification in 2025 for both CPU and MCU architectures. Complementary solutions address additional security needs, including hypervisor-based threat containment and TrustZone-embedded protection for ECUs.

What differentiates Perseus from other hypervisor providers?

Perseus combines more than 18 years of embedded virtualization expertise with open-source leadership and production deployment at the foundational system software layer. The company contributes to industry standards through its leadership role in COVESA (formerly GENIVI), focusing on long-term interoperability and platform sustainability rather than proprietary lock- in.

Boilerplate

Perseus is an automotive system software company specializing in foundational system software for Software-Defined Vehicle (SDV) architectures. With more than 18 years of experience in embedded systems and virtualization R&D, Perseus develops system-level software that operates below the operating system layer—where efficiency, security, and performance are determined.

Perseus' portfolio includes the ISO 26262 ASIL-D certified PEGASUS automotive hypervisor for CPU and MCU architectures, hypervisor-based security solutions, Linux fast-boot technology, MCU secure boot software, and supporting development and operational tooling. Together, these solutions enable the safe and deterministic execution of multiple operating systems and workloads on shared hardware, addressing the growing complexity of modern, software-centric vehicle platforms.

Perseus works with automotive OEMs, Tier-1 and Tier-2 suppliers, semiconductor vendors, and ecosystem partners developing SDV platforms. Its software is designed to integrate with existing automotive hardware and operating systems, supporting long vehicle lifecycles and production deployment.

Perseus' credibility is demonstrated through functional safety certification, production programs, semiconductor ecosystem alignment, open-source leadership, and collaboration with leading research institutions.

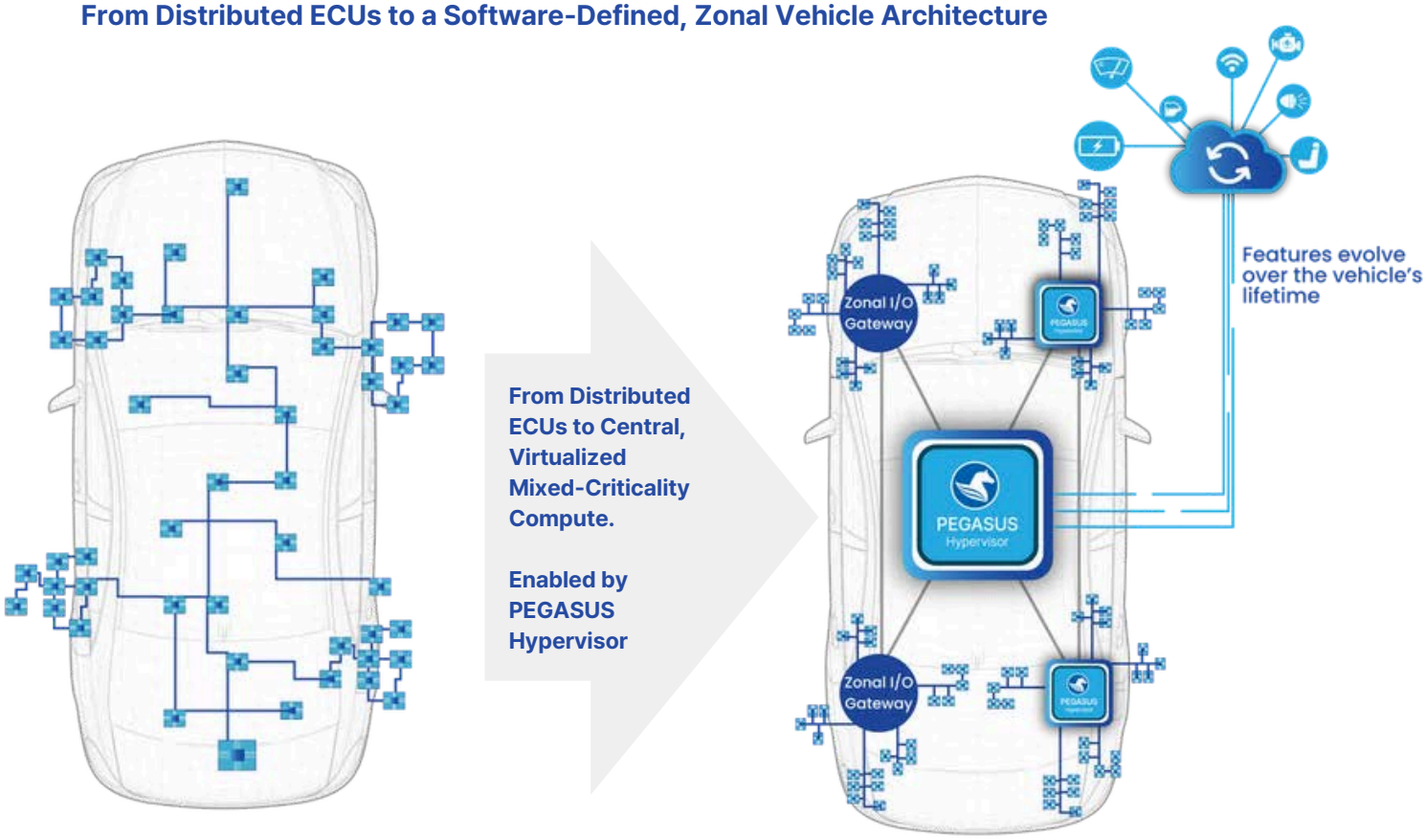
More Information

www.cyberperseus.com
sales@cyberperseus.com

Download our solution brochures

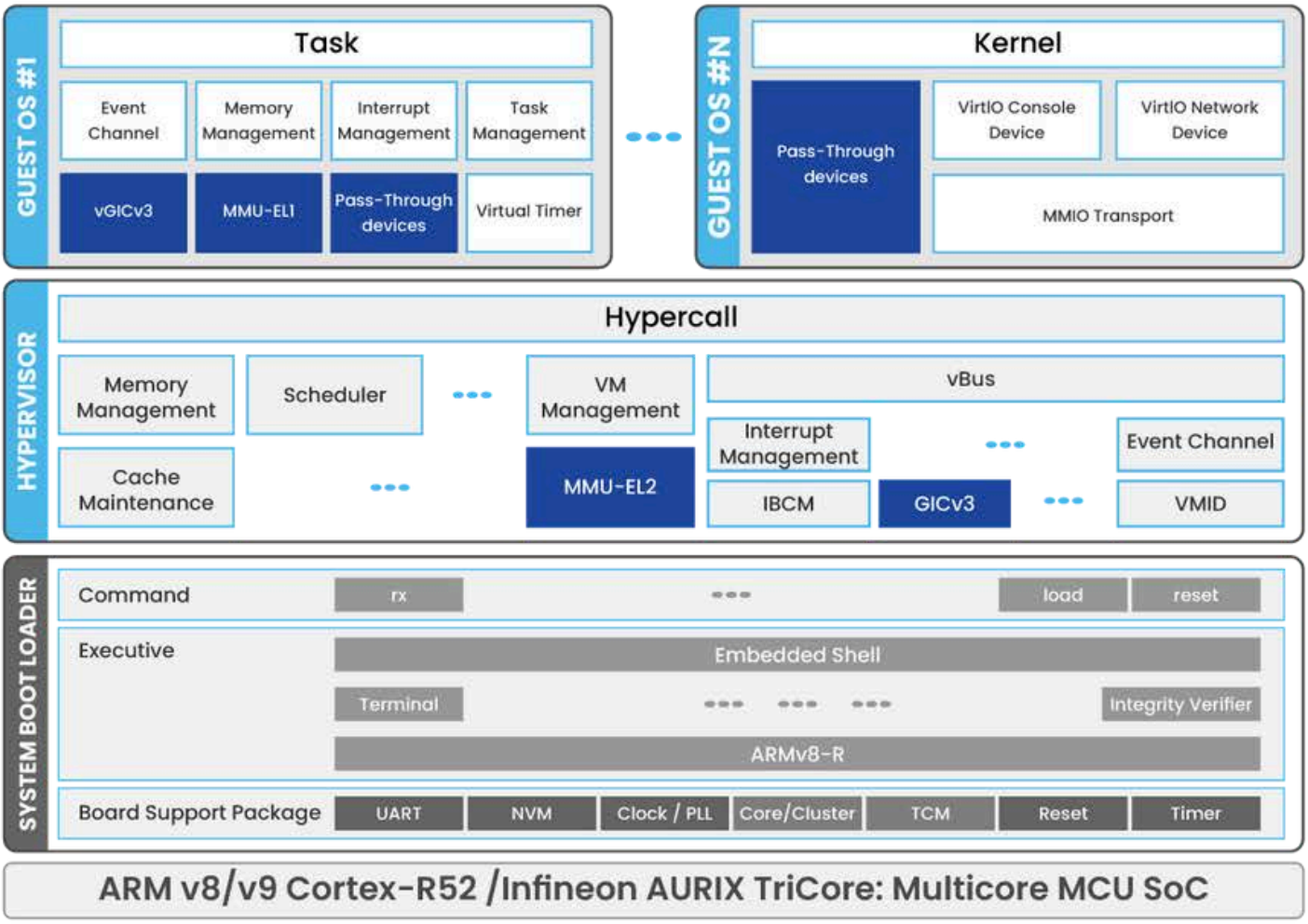
APPENDIX

From Distributed ECUs to a Software-Defined, Zonal Vehicle Architecture



In-vehicle E/E architectures are transitioning from traditional, signal-centric ECU architecture (left) to a zonal, software-defined vehicle architecture (right). Perseus' PEGASUS Hypervisor consolidates mixed-criticality workloads onto centralized compute, abstracting hardware and enforcing strong isolation, determinism, and resource control. Zonal I/O gateways aggregate local sensors and actuators, reducing wiring complexity and decoupling hardware from software functions. The hypervisor enables multiple operating systems and domains to run concurrently on shared SoCs, supporting scalable function deployment, including OTA management and updates and deterministic behavior - key enablers of modern SDV platforms.

PEGASUS Hypervisor on MCUs: Deterministic, Mixed-Criticality Control for Real-Time, Safety-Constrained SDV Platforms



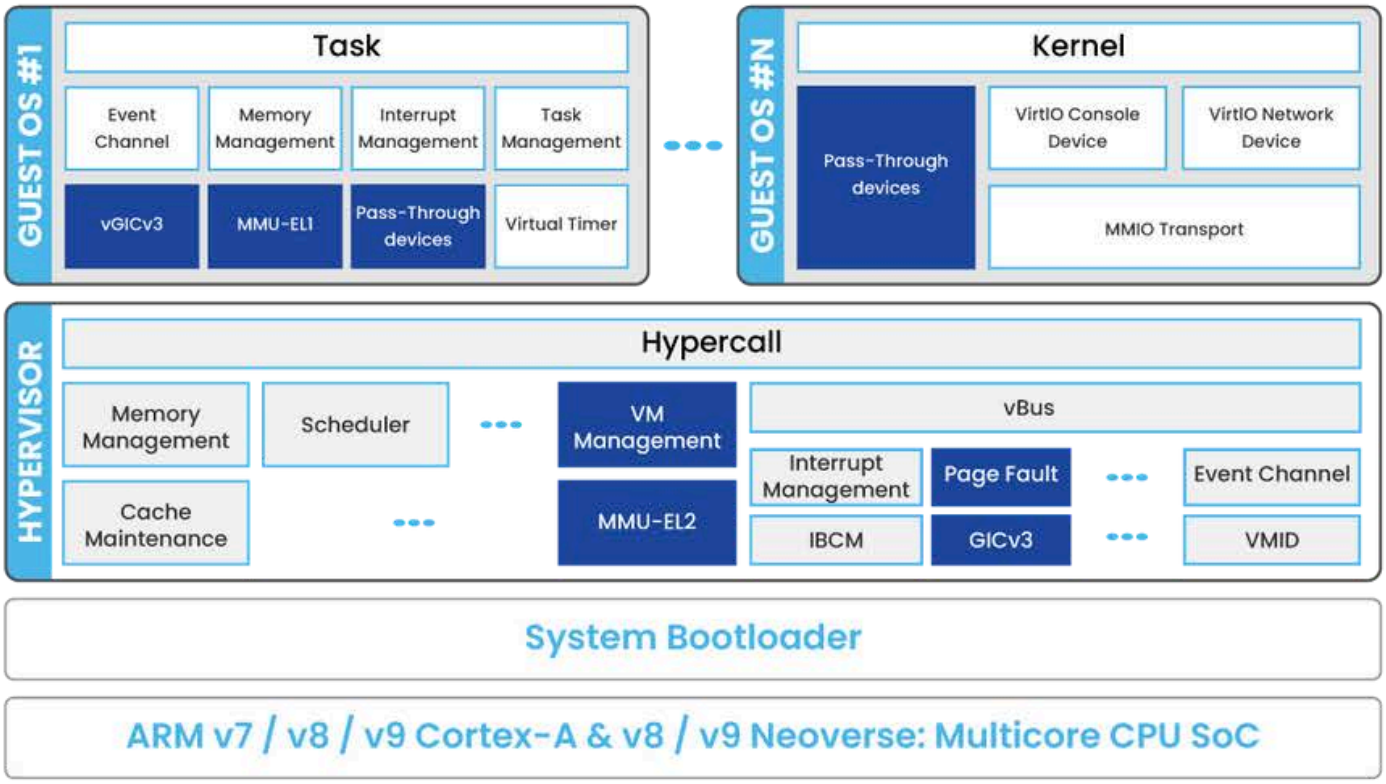
<MCU-based Hypervisor System Architecture>

This diagram highlights the role of the **Perseus PEGASUS hypervisor** as the control and isolation layer in an MCU-based SDV architecture. The hypervisor manages scheduling, memory, interrupts, and VM lifecycles, enforcing deterministic execution across mixed-criticality workloads.

Dark blue blocks in the graphic indicate hardware-assisted functions or tightly controlled pass- through paths—such as MMU-EL2, GICv3, and interrupt routing—that anchor real-time behavior and strong isolation. In the guest OS layer, dark blue components represent performance-critical or directly mapped devices, balancing virtualization with low-latency access. Together, these mechanisms enable predictable timing, secure partitioning, and efficient resource sharing on constrained multicore MCUs.

GAIA, Perseus' secure bootloader, is an optional component that complements the PEGASUS hypervisor by providing a secure, MCU-agnostic boot layer. GAIA initializes hardware, verifies system integrity, and establishes a consistent execution environment before virtualization. By abstracting board-specific details and enforcing secure boot early in the startup sequence, GAIA enables predictable bring-up and broad hardware compatibility across diverse MCU platforms.

PEGASUS Hypervisor on CPUs: Scalable Virtualization for Consolidated, Mixed-Criticality SDV Compute on High-Performance SoCs



Building on the same isolation, determinism, and mixed-criticality principles established in the MCU architecture, the Perseus PEGASUS hypervisor scales these capabilities to CPU-based platforms, enabling consolidation of richer operating systems and higher-performance workloads on centralized SDV compute.

Where the MCU-based architecture prioritizes tight control on resource-constrained, real-time cores, the CPU-based architecture extends the same foundations to higher-performance, feature-rich SoCs. On Cortex-A platforms, the PEGASUS hypervisor manages larger and more complex guest operating systems, richer device models, and deeper levels of virtualization.

Additional mechanisms—such as VirtIO, vBus, page fault handling, and MMIO transport— support scalable I/O virtualization and dynamic resource sharing. While determinism, isolation, and mixed-criticality enforcement remain consistent, the CPU architecture emphasizes flexibility, throughput, and OS consolidation, enabling Linux-class workloads alongside real- time domains within a unified, software-defined vehicle compute platform.