Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VI, v.2 2025 | Submission: 02/08/2025 | Accepted: 05/08/2025 | Publication: 08/08/2025

Multiplatform software architecture, embedded systems, and edge computing: an analysis of hardware and software integration in highly complex environments.

Cross-platform software architecture, embedded systems, and edge computing: an analysis of hardware-software integration in high-complexity environments

Multiplatform software architecture, embedded systems and edge computing: an analysis of integration between hardware and software in highly complex environments

Matheus Piotroski Neis

Computer Engineer graduated from the Federal University of Santa Maria (UFSM), Electronics Technician from SENAI, and Master in Administration, Finance and Value Creation from PUCRS.

Summary

This article analyzes the principles of software engineering and systems architecture applied to the development of multiplatform solutions and the integration of embedded systems in high-demand corporate environments. Based on the literature of computing, systems engineering, and data from the technology sector, the study investigates the bottlenecks of latency, interoperability, and maintainability in the transition between legacy systems and modern Edge Computing infrastructures. The application of design patterns , low- and high-level programming languages, and agile development methodologies as vectors to ensure scalability and operational efficiency in digital platforms and physical devices is discussed.

Keywords: Software engineering. Embedded systems. Edge computing. Interoperability. Multiplatform architecture.

Abstract

This article analyzes software engineering principles and systems architecture applied to the development of cross-platform solutions and embedded systems integration in high-demand corporate environments. Grounded in computer science literature, systems engineering, and technology sector data, the study investigates latency, interoperability, and maintainability bottlenecks during the transition from legacy systems to modern Edge Computing infrastructures. It discusses the application of design patterns, low- and high-level programming languages, and agile development methodologies as drivers to ensure scalability and operational efficiency across digital platforms and physical devices.

Keywords: Software engineering. Embedded systems. Edge Computing. Interoperability. Cross-platform architecture.

Introduction

The accelerated evolution of global digital infrastructure demands from the field of computer engineering...

Development of increasingly integrated, resilient, and scalable systems. As conceptualized

According to Pressman (2014), contemporary software engineering transcends the simple act of writing code.

establishing itself as a rigorous discipline that covers the complete life cycle of

development, from requirements analysis to systems architecture and maintenance

Continued. In the industrial and corporate context, the proliferation of Internet of Things devices.

(IoT) and the need for local processing have created an unprecedented demand for architectures

capable of seamlessly integrating cloud-based software modules with embedded firmware.

Limited hardware.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VI, v.2 2025 | Submission: 02/08/2025 | Accepted: 05/08/2025 | Publication: 08/08/2025

The fragmentation of operating systems, such as Windows, macOS, Linux, iOS, and Android, establishes a significant technical challenge for enterprise software client development. The search through cross-platform interoperability without loss of performance or compromise of User experience requires the use of efficient languages and the proper abstraction of layers. Hardware. Architectural inadequacy in the software design phase results in high costs. refactoring, performance vulnerabilities, and extreme difficulties for scalability of operations in large-scale environments.

This article aims to discuss best practices in software architecture and engineering. embedded systems, relying on established theoretical frameworks in computer science. Through the systematic analysis of the principles of Edge Computing, circuit integration and In cross-platform development, the aim is to demonstrate that the rigorous application of standards... Design and alignment between hardware and software are the determining factors for efficiency. operational efficiency, latency reduction, and longevity of complex technological platforms.

Methodology

The methodological approach adopted in this study is based on a systematic literature review and on... descriptive analysis of software engineering models and computer architecture, a framework based on Sommerville (2019) for the investigation of complex systems. Data collection It is based on academic literature in computer engineering, technical specifications of microelectronics, distributed network performance reports, and standards documentation Industrial cybersecurity.

The analysis includes system performance metrics such as response time, network latency, memory consumption, data transfer rate *(throughput)* , and energy efficiency in embedded processors. These variables were correlated with theoretical frameworks. developed by authors such as Tanenbaum and Bos (2015), Patterson and Hennessy (2017) and Lee and Seshia (2017), who investigate the interaction between operating systems, hardware architecture and cyber- physical systems .

Additionally, the methodology evaluates the effectiveness of modern architectural standards, such as Microservices, asynchronous communication, and event-driven architectures in Edge Computing. The impact of adopting continuous integration and continuous delivery (CI/CD) practices combined with... is examined. Agile development methodologies, validating the rigor in the design and testing phase. Automated systems prevent catastrophic failures in mission-critical systems.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS - Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VI, v.2 2025 | Submission: 02/08/2025 | Accepted: 05/08/2025 | Publication: 08/08/2025

Chapter 1: Cross-Platform Software Architecture and Interoperability Challenges

The development of enterprise software designed to operate across multiple operating systems and Processor architectures require rigorous architectural planning to ensure consistency. functional and system performance. According to Sommerville (2019), software construction Cross-platform development should minimize the coupling between application code and specific APIs. each operating system. The proper abstraction of the operating system access layers. It allows business logic to remain agnostic with respect to the underlying platform, making it easier to... The portability and maintenance of the code over time.

The choice of development languages and frameworks is a critical decision in Systems engineering. Languages like C and C++ remain the gold standard for modules that require high performance and direct access to system memory, while languages How Python and JavaScript offer agility in developing cloud services and interfaces. of the user. The harmonious integration between these different technological layers requires the definition clear programming interfaces and the use of efficient communication protocols to avoid processing bottlenecks.

Managing application state and synchronizing data between local clients and servers in Cloud computing represents a significant operational bottleneck in multi-platform architectures. According to Tanenbaum and Van Steen (2017) analyze fault tolerance and... Data consistency should be designed considering the possibility of temporary disconnections. network. The implementation of local caching mechanisms with strategies of Data reconciliation upon reconnection ensures application usability in high-demand scenarios. connectivity instability.

Furthermore, quality control of cross-platform software requires comprehensive automation of Regression, integration, and performance testing in CI/CD pipelines. Execution of test suites. automated in virtualized environments representing different hardware configurations and The operating system ensures the early identification of compatibility issues. This rigor A methodological approach in the testing cycle drastically reduces the occurrence of errors in a production environment. preserving the reliability of the software in the eyes of corporate users.

It can be concluded that high-quality cross-platform software engineering depends on balance. between architectural abstraction, code efficiency, and rigor in testing processes. The adoption of patterns. Universal design principles and the isolation of platform-specific features ensure scalability. of the technological solution, enabling corporations to operate complex digital environments with costs Reduced maintenance and high level of operational stability.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS - Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VI, v.2 2025 | Submission: 02/08/2025 | Accepted: 05/08/2025 | Publication: 08/08/2025

Chapter 2: Embedded Systems Engineering and Edge Computing

The transition from purely cloud-centric processing to decentralized architectures of
Edge computing has redefined the role of embedded systems in the industrial and corporate ecosystem.
According to Lee and Seshia (2017), embedded systems are designed to perform
specific tasks under severe constraints on response time, energy consumption, and capacity.
memory. Transferring computational intelligence to the network edge reduces dependence on
It increases bandwidth and reduces latency in decision-making in automated mission processes.
Criticism.

Embedded firmware and software design requires a deep understanding of the architecture of...
microcontrollers and microprocessors used on the device's circuit board. Programming in
Low-level programming, using C or Assembly, allows for precise manipulation of registers and interrupts.
of hardware and communication peripherals such as UART, SPI, and I2C. According to Patterson and Hennessy
(2017), optimizing RAM usage and processor cycle time is fundamental
to avoid resource depletion in devices that need to operate continuously in
hard-to-reach environments.

The integration of analog and digital sensors into Internet of Things (IoT) platforms demands
The proper handling of physical signals and high-precision analog-to-digital conversion. The design of
Electronic circuits and printed circuit board (PCB) layouts must provide for filtering of
Electromagnetic noise and protection against voltage variations to ensure data integrity.
collected by the sensors. The reliability of the embedded software is intrinsically linked to
stability and quality of the hardware on which it runs.
Remote firmware updates *(Over-The-Air* - OTA) in distributed fleets of devices.
Embedded systems represent one of the greatest safety and engineering challenges in the sector.
Implementing secure update mechanisms requires verification of the digital signature of...
Software image, the use of dual memory partitions to avoid irreversible crashing of the
device *(bricking)* in case of failure during download, and encryption of the channels of
communication. The robustness of this process is essential for maintaining the operational safety of
smart physical infrastructures
Therefore, embedded systems engineering in Edge Computing demands a holistic view.
which includes hardware design, efficient firmware development, and security of
Network connectivity. The perfect synergy between the electronics and the control software allows for the creation...
resilient autonomous solutions, prepared to support the processing demands in
Real-time of modern industry.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VI, v.2 2025 | Submission: 02/08/2025 | Accepted: 05/08/2025 | Publication: 08/08/2025

Chapter 3: Critical Infrastructure Management and Data Center Automation

The operation of data centers and critical IT infrastructures requires the implementation of systems.
Continuous management and monitoring with very high levels of availability and security.
As analyzed by Tanenbaum and Bos (2015), the management of servers and network assets in
Large-scale operations are based on *out-of-band access management,* allowing that
System administrators can access, diagnose, and reconfigure hardware remotely, even
when the main network or the server's operating system is inoperative.
Automating data center management relies on the continuous collection of environmental telemetry.
and operational via sensors for temperature, humidity, energy consumption, and port status.
From cabinets. The aggregation of this data by concentrator devices and its secure transmission.
Cloud-based platforms enable predictive analysis of failures in refrigeration equipment.
and power supplies. Early detection of operational anomalies prevents unforeseen interruptions.
planned *downtime,* the costs of which are extremely high for large corporations.
Developing embedded software for data center management devices requires...
Implementation of rigorous industrial and corporate network protocols, such as SNMP, Modbus,
Redfish and SSH. The firmware for these devices must be designed to ensure proper execution.
Simultaneous monitoring and control tasks through real-time operating systems.
(RTOS) or optimized embedded Linux distributions. Process isolation and strict management
Permissions prevent the compromise of a secondary service from affecting the control of
critical device functions.
Physical and digital security in infrastructure management requires the integration of mechanisms.
Strong authentication and immutable auditing of executed commands. Centralization of records of
access to encrypted logs and the implementation of role-based access control (RBAC)
They ensure that only authorized professionals perform configuration changes on the infrastructure.
Protection against denial-of-service (DoS) attacks and port scanning on devices.
Out-of-band management is imperative to maintain data center resilience.
It can be concluded that automation and secure data center management represent the fusion between
Hardware engineering, computer networks, and control software. Continuous modernization.
these platforms through robust embedded software and management interfaces.
Centralized cloud solutions ensure energy efficiency, structural security, and maximum security.
Availability of the digital infrastructures that underpin the global economy.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS - Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VI, v.2 2025 | Submission: 02/08/2025 | Accepted: 05/08/2025 | Publication: 08/08/2025

Chapter 4: Structural Cybersecurity, Zero Trust Principles, and Global Network Protection

The increasing sophistication of cyberattacks on a global scale has rendered previous models obsolete.
Traditional perimeter security approaches are based on the concept of a "trusted network perimeter." According to...
According to Stallings (2017), the Zero Trust architecture redefines digital security under the principle...
The fundamental principle is "never trust, always verify." In this paradigm, the user's identity and status are paramount.
The device's security and the context of the request must be authenticated and authorized.
continuously, regardless of whether the connection originates internally or outside the network.
corporate.
Implementing a Zero Trust architecture for accessing global networks requires the development of...
of software clients capable of establishing highly secure and dynamic communication tunnels.
The use of end-to-end encryption protocols, coupled with the verification of digital certificates.
and multi-factor authentication (MFA) tokens, ensure the confidentiality and integrity of
Data traffic. Network microsegmentation is applied to restrict user access.
strictly limited to the resources necessary for the performance of their duties, restricting movement.
wing of potential attackers.
Protecting software clients at the endpoint level requires the integration of defense mechanisms.
Assets against reverse engineering, code injection, and exploitation of memory vulnerabilities. A
use of code obfuscation techniques , integrity verification of
executables at runtime and the correct handling of cryptographic keys in modules of
Hardware-based security (such as TPM - *Trusted Platform Module)* increases
significantly improves the application's resistance against malicious analyses.
Vulnerability management in the software development lifecycle *(Secure SDLC)* is a
A mandatory requirement for ensuring structural cybersecurity. The systematic performance of analyses.
Static and dynamic code analysis (SAST and DAST), along with peer reviews and penetration testing.
(penetration testing) allows you to identify and correct security flaws before the release of
application. The rapid application of security patches and updates in response to the discovery of
Emerging global threats preserve the organization's security posture.
In short, cybersecurity cannot be treated as a layer added after the fact.
not just software, but as a fundamental design requirement from the architectural phase onwards. The implementation
strict adherence to Zero Trust principles and advanced protection of global networks through software.
Resilient customers ensure business continuity, the protection of sensitive data, and...
preserving the digital sovereignty of organizations in the face of cyber threats in
constant evolution.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VI, v.2 2025 | Submission: 02/08/2025 | Accepted: 05/08/2025 | Publication: 08/08/2025

Chapter 5: Application of Artificial Intelligence and Automation in Optimization of Software Development

The convergence between enterprise artificial intelligence and machine learning *(Machine Learning)* and software engineering revolutionized automation and development processes. complex systems. According to Russell and Norvig (2021), the application of intelligent algorithms for the analysis of operational data and source code, it allows the identification of patterns and the prediction of Failure management and autonomous optimization of performance parameters. In modern software engineering, AI is no longer just a product feature, but a support tool for... development cycle.

Intelligent automation in code review and generation significantly accelerates the processes of Quality assurance and architectural compliance. Analytical tools based on models of Advanced programming languages can examine code repositories in search of anti-patterns and bottlenecks. memory and deviations from company security guidelines. Automated identification of Identifying logical weaknesses before compilation optimizes software engineers' time, allowing them to... Focus on highly complex tasks and systems design.

In the field of software operations and cloud monitoring (AIOps), artificial intelligence is... Used to analyze massive volumes of logs, network metrics, and real-time telemetry records. Real. Anomaly detection algorithms can identify subtle deviations in the behavior of systems that anticipate serious hardware failures or cyberattacks, triggering routines Automated remediation strategies, such as reactivating microservices or redirecting traffic. network, before the end user is impacted.

Applying AI at the network edge (Edge AI) requires optimizing machine learning models. A machine designed for execution on embedded devices with limited computing power. Model quantization and neural network pruning techniques *reduce* the memory footprint. and the processing demands of AI, enabling microcontrollers and IoT devices perform tasks involving computer vision, natural language processing, and sensory analysis. Locally, with very low latency and without the need for a constant connection to the cloud.

It is concluded that the strategic integration of artificial intelligence in software engineering and in Embedded systems have redefined the standards for productivity, safety, and operational efficiency. The responsible application of intelligent algorithms enhances the analytical capabilities of engineers. It optimizes the performance of real-time systems and paves the way for the creation of solutions. Autonomous and highly resilient, prepared for the future of the digital industry.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VI, v.2 2025 | Submission: 02/08/2025 | Accepted: 05/08/2025 | Publication: 08/08/2025

Final Considerations

The analysis presented throughout this article highlights the evolution of computer engineering.
Modern architecture is based on the harmonious convergence of cross-platform software architecture.
Efficient embedded systems and advanced structural cybersecurity practices. The research
This demonstrates that the increasing complexity of digital ecosystems requires that the engineer...
The software possesses a deep systemic view, capable of connecting the physical constraints of the circuits.
electronics and microprocessors to the abstract layers of cloud services and user applications.
end.

Overcoming interoperability bottlenecks in cross-platform software clients and...
Accelerating the transition to edge computing infrastructures depends strictly on adoption.
Disciplined design patterns and appropriate operating system abstractions. Planning
A rigorous architectural approach in the initial software design phase significantly reduces costs.
Ongoing maintenance eliminates memory vulnerabilities and prevents performance loss.
resulting from execution on different processor architectures.
The study reinforces the centrality of edge processing for the success of the Internet of Things.
industrial and for the autonomous management of critical infrastructures, such as data centers.
Low-level firmware development optimized for limited hardware resources ensures
precise telemetry collection and real-time command execution, while mechanisms
Robust OTA remote update and communication protection ensure longevity and...
Physical and digital security of distributed device fleets.
Furthermore, the mandatory incorporation of cybersecurity principles based on the Zero Trust model.
It establishes an indispensable defensive barrier against the proliferation of global digital threats.
Client software for accessing corporate networks should be designed according to authentication principles.
Continuous, end-to-end encryption and active endpoint defense, ensuring connectivity.
Do not let multiplatform become a vulnerability vector for the infrastructure of
organization.
The proactive application of artificial intelligence in the software development lifecycle and in
System monitoring (AIOps and Edge AI) emerges as the new paradigm for scaling.
and productivity. Analytical code automation and predictive remediation of operational failures.
They expand the engineering capacity to deliver mission-critical systems with extremely high performance.
availability, while optimized AI models bring autonomy.
Computational capabilities for network edge devices.
The systematization of these engineering methodologies into technical manuals and reference architectures.

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Machine Translated by Google
RCMOS – Multidisciplinary Scientific Journal The Knowledge.
ISSN: 2675-9128. São Paulo-SP.

Year VI, v.2 2025 | Submission: 02/08/2025 | Accepted: 05/08/2025 | Publication: 08/08/2025

Instructional materials act as a powerful catalyst for raising the maturity of the sector.
technological. The continuous transfer of this specialized knowledge to new professionals and
Corporate teams strengthen organizations' capacity for innovation and prevent design failures.
which result in wasted capital and national security risks.

Given the scenario of accelerated digital transformation and the convergence between the physical and digital worlds...
In the digital age, rigorous software engineering is becoming the backbone of the global economy.
investment in clean architectural design, in optimizing the integration between hardware and software and
Continuous technical training is the only way to create technological platforms.
Sustainable, safe, and truly transformative.

It can be concluded that academic and technical rigor in the application of computer science is the key factor.
The determining factor that differentiates ephemeral solutions from robust and scalable digital platforms.
A perfect integration between theory and practice, supported by the relentless pursuit of simplicity.
Architectural and structural safety considerations are what allow computer engineering to build the future.
of global digital infrastructure with excellence, resilience and high productive impact.

References

- LEE, EA; SESHIA, SA *Introduction to Embedded Systems: A Cyber-Physical Systems Approach.* 2nd ed. Cambridge: MIT Press, 2017.
- PATTERSON, DA; HENNESSY, JL *Computer Organization and Design: The Hardware/ Software Interface.* 5. ed. Cambridge: Morgan Kaufmann, 2017.
- PRESSMAN, RS *Software Engineering: A Professional Approach.* 8th ed. Porto Alegre: AMGH, 2014.
- RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach.* 4th ed. Hoboken: Pearson, 2021.
- SOMMERVILLE, I. *Software Engineering.* 10th ed. São Paulo: Pearson, 2019.
- STALLINGS, W. *Cryptography and Network Security: Principles and Practice.* 7. ed. Boston: Pearson, 2017.
- TANENBAUM, AS; BOS, H. *Modern Operating Systems.* 4th ed. São Paulo: Pearson, 2015.
- TANENBAUM, AS; VAN STEEN, M. *Distributed Systems: Principles and Paradigms.* 3rd ed. Delft: CreateSpace, 2017

This article is published in open access under the Creative Commons Attribution license, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.