

Review

A Scoping Review of Composite Evaluation Approaches for Software Architectures on Resource-Constrained Hardware

Alejandro Israel López Cortes ^{1,*}, José Ricardo Gómez Rodríguez ^{1,*}, Cristian Eduardo Boyain y Goytia Luna ¹, Abraham Jair López Villalvazo ², Remberto Sandoval Aréchiga ¹ and Viktor Iván Rodríguez Abdalá ¹

¹ Posgrado en Ingeniería para la Innovación Tecnológica, Unidad Académica de Ingeniería Eléctrica, Universidad Autónoma de Zacatecas “Francisco García Salinas”, Campus Siglo XXI, Carretera Zacatecas-Guadalajara Km 6, Zacatecas 98160, Zacatecas, Mexico; cristian.boyain@uaz.edu.mx (C.E.B.y.G.L.); rsandoval@uaz.edu.mx (R.S.A.); abdala@uaz.edu.mx (V.I.R.A.)

² Centro Universitario del Sur, Universidad de Guadalajara, Ciudad Guzmán 49000, Jalisco, Mexico; abraham@cusur.udg.mx

* Correspondence: israel.lopez@uaz.edu.mx (A.I.L.C.); jrgrodri@uaz.edu.mx (J.R.G.R.)

Featured Application

This scoping review supports the selection and design of composite evaluation approaches for architecture-relevant software or system artifacts deployed on resource-constrained hardware platforms.

Abstract

Background: Hardware-software co-design for resource-constrained systems requires evaluation methods that integrate multiple quality attributes rather than isolated performance indicators. **Objectives:** This scoping review maps composite evaluation approaches used to assess architecture-relevant software or system artifacts, including software architectures where applicable, on resource-constrained hardware, while distinguishing explicitly aggregated quantitative composite metrics from qualitative and user-centric forms of evaluation. **Methods:** Using a PRISMA-ScR-informed reporting structure, the review analyzes a curated corpus of publications to identify foundational quality attributes, evaluation mechanisms, target hardware platforms, and architectural trade-off methods. **Title and abstract screening, full-text eligibility assessment, and data charting were conducted by two investigators, with a third investigator available to resolve discrepancies when necessary.** **Results:** Within the reviewed corpus, a descriptive shift is observed from single-attribute optimization toward broader multi-attribute evaluation, including explicit objective functions, benchmark-guided system-level modeling, structured qualitative trade-off assessment, and Quality-of-Experience (QoE)-oriented evaluation. These mechanisms are reported across FPGA-based accelerators, ARM-based edge devices, low-end IoT frameworks, heterogeneous intelligent devices, and XR platforms. However, explicitly aggregated architectural-level composite metrics remain rare, and representative studies such as FLASH, HW-FlowQ, Monarch, and ILLIXR illustrate related but distinct evaluation forms, rather than a single unified metric family. **Conclusions:** The reviewed corpus indicates that composite evaluation approaches support architecture-relevant reasoning under hardware constraints, although further work is needed to improve reproducibility, operational consistency, and cross-domain validation.



Academic Editor: Jose María Alvarez Rodríguez

Received: 12 May 2026

Revised: 15 June 2026

Accepted: 23 June 2026

Published: 28 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: composite evaluation approaches; composite metrics; software architecture; hardware-software co-design; quality attributes; performance evaluation; resource-constrained hardware; scoping review; technical debt; neural architecture search; quality-of-experience (QoE)

1. Introduction

The decline of conventional CMOS scaling has fundamentally altered the landscape of computing, driving a pronounced shift towards specialized, heterogeneous systems-on-chip (SoCs) [1–4]. This era of domain-specific computing is particularly evident in resource-constrained edge devices deployed in fields like Extended Reality (XR), the Internet of Things (IoT), and autonomous systems. For these platforms, where power, latency, and area are critically constrained, this complexity necessitates composite evaluation frameworks that integrate multiple trade-offs, such as those implemented in Monarch [5] for hyperscale telemetry, and in ILLIXR [4] for immersive XR platforms. The traditional, siloed evaluation of software performance is no longer sufficient. Achieving desired system-level outcomes requires a deeply integrated approach to system design.

This paradigm shift has elevated hardware-software (HW/SW) co-design from a niche discipline to a critical, mainstream methodology [6–11]. In a co-design context, the software architecture and the underlying hardware are designed and optimized in tandem, forcing developers to manage complex trade-offs between often conflicting system goals. For example, an architectural choice that improves raw performance might lead to unacceptable energy consumption, while a design optimized for maintainability could introduce prohibitive latency. Evaluating these trade-offs requires metrics that can capture a holistic view of system quality.

In this manuscript, the term *resource-constrained hardware* is used as the main umbrella term for embedded, edge, and specialized platforms operating under explicit limits on power, latency, area, memory, or computing capacity. This includes not only classical embedded and IoT devices, but also ARM-based edge platforms, batteryless or energy-harvesting devices, FPGA/eFPGA accelerators, heterogeneous edge nodes, and domain-specific platforms whose evaluation is governed by tight resource or real-time constraints.

Accordingly, the primary objective of this scoping review is to map the landscape of composite evaluation approaches used for the holistic evaluation of architecture-relevant software or system artifacts designed for resource-constrained hardware, including software architectures where the primary studies report them explicitly. Within this broader family, the term *composite metrics* is reserved for explicitly aggregated quantitative formulations that combine multiple attributes through a defined operational expression, whereas qualitative trade-off frameworks and user-centric QoE-oriented evaluations are treated as related but distinct evaluation approaches. These approaches move beyond measuring individual characteristics like throughput or coupling and instead provide a structured basis for reasoning about the overall fitness of an architectural solution in its target hardware environment.

This report is structured as follows. Section 2 details the scoping review methodology employed to gather and analyze the findings. Section 3 presents the results, starting with foundational single-dimension quality attributes and progressing to the various forms of composite evaluation approaches identified in the reviewed corpus. Section 4 provides a discussion of the strategic implications of these findings, including their impact on development practices and the concept of architectural technical debt. Finally, Section 5 offers a conclusion and outlines promising directions for future work in this domain.

2. Materials and Methods

This review was designed as a scoping review and structured using the reporting categories of the PRISMA extension for Scoping Reviews (PRISMA-ScR). A scoping review approach was selected because the objective was to map how composite metrics and related composite evaluation approaches are defined, combined, and applied across heterogeneous hardware-software co-design contexts, rather than to estimate the effect of a single intervention. Title and abstract screening, full-text eligibility assessment, and data charting were conducted by two investigators, with a third investigator available to resolve discrepancies when necessary. The review should therefore be interpreted as a structured scoping synthesis with acknowledged methodological limitations, but not as a purely single-reviewer workflow.

2.1. Research Questions

The review was guided by a set of core research questions designed to investigate the landscape of architectural evaluation for resource-constrained systems. These questions are as follows:

- RQ1: Which foundational quality attributes and individual metrics are used to evaluate software architecture in the context of resource-constrained hardware?
- RQ2: How are these individual metrics combined or synthesized into explicit composite metrics, qualitative trade-off frameworks, or user-centric evaluation approaches to support trade-off analysis?
- RQ3: What architecture-relevant software or system artifact classes and hardware platforms are most prominently discussed in recent HW/SW co-design literature?

2.2. Protocol and Registration

A protocol for this scoping review was registered in OSF Registries (DOI: <https://doi.org/10.17605/OSF.IO/U85BV>) and is available at <https://osf.io/u85bv/> (accessed on 14 June 2026). The review questions, eligibility criteria, screening procedure, and data-charting categories were defined before full-text analysis and refined iteratively as the terminology used in the literature was mapped.

2.3. Information Sources and Search Strategy

The evidence base consisted of a curated corpus of peer-reviewed publications and research papers addressing software architecture evaluation, quality attributes, performance metrics, hardware-software co-design, edge computing, embedded systems, and specialized hardware platforms. The primary information sources used to identify candidate records were Web of Science, Scopus, and Google Scholar. Searches were conducted from December 2025 to January 2026, prior to submission of the present manuscript version in May 2026. Because the literature uses heterogeneous terminology, the corpus was screened using both metric-centered and broader trade-off concepts represented in the review questions: composite metrics, software architecture, quality attributes, hardware-software co-design, resource-constrained hardware, embedded systems, specialized accelerators, edge devices, latency, energy, performance, maintainability, and quality of experience.

The search logic used to identify and verify topical relevance can be represented as follows: ("software architecture" OR "software architectures") AND ("composite metric" OR "composite metrics" OR "quality attribute" OR "quality attributes" OR "multi-objective" OR "trade-off") AND ("hardware-software co-design" OR "embedded system" OR "edge computing" OR "resource-constrained" OR "specialized hardware" OR accelerator) AND (performance OR latency OR energy OR maintainability OR reliability OR portability OR

“quality of experience”). This conceptual strategy was adapted to the syntax and filtering options available in each information source.

The records were organized and screened according to PRISMA-ScR principles. The initial identification stage yielded 512 records. After removing 120 duplicate entries, 392 unique records remained for title and abstract screening. A total of 188 records were excluded during screening because they did not explicitly address hardware constraints, software architecture evaluation, or multi-attribute evaluation approaches. The remaining 204 articles were assessed through full-text analysis. Finally, 38 articles were excluded because they did not meet the inclusion criteria related to trade-off analysis or metric formalization, resulting in 166 primary studies included in the final synthesis. The source-selection process is summarized in Figure 1.



Figure 1. PRISMA-ScR flow diagram illustrating the source selection process, including identification, screening, eligibility, and inclusion phases.

2.4. Eligibility Criteria

Sources were considered eligible when they described metrics, quality attributes, evaluation frameworks, or trade-off methods relevant to software architectures operating on resource-constrained hardware platforms. This included studies that either directly evaluated software architectures or presented hardware-aware optimization, co-design, or benchmark-driven analyses that materially informed architecture-level trade-off reasoning under hardware constraints. The review focused on sources published between 2015 and 2025 with sufficient English-language metadata and methodological detail for data charting. Studies were excluded when they focused only on software quality without hardware or deployment constraints, lacked a concrete metric or evaluation method, or did not support analysis of trade-offs among multiple system characteristics.

The inclusion and exclusion criteria are summarized in Table 1.

Table 1. Eligibility criteria used for source selection.

Code	Criterion
I1	Include studies describing quality attributes, metrics, or evaluation frameworks for software architectures or hardware/software co-designed systems with architecture-relevant implications.
I2	Include studies targeting resource-constrained embedded, edge, or specialized hardware platforms.
I3	Include studies involving trade-off analysis between multiple system characteristics, such as performance, energy, latency, maintainability, reliability, portability, resource efficiency, or quality of experience.
E1	Exclude studies focused purely on software without explicit consideration of hardware constraints, deployment context, or performance-resource trade-offs.
E2	Exclude studies that do not provide concrete metrics, measurable quality attributes, or a structured evaluation methodology.
E3	Exclude studies for which the full text or sufficient methodological information was not available for charting.

2.5. Selection of Sources of Evidence

Titles and abstracts were screened first to remove records outside the scope of the review. Full texts were then assessed against the eligibility criteria. The selection process prioritized evidence that explicitly connected software architecture decisions, or architecture-relevant system design choices, with hardware constraints and multi-attribute evaluation. Reasons for exclusion were recorded at the screening and full-text eligibility stages and summarized in the PRISMA-ScR flow diagram. Screening and eligibility assessment were conducted by two investigators, and any discrepancies were resolved through consultation with a third investigator. This procedure improves the robustness of study selection relative to a purely single-reviewer workflow, although it does not eliminate all sources of selection bias.

2.6. Data Charting and Data Items

The selected source documents were charted using a structured form derived from the research questions and eligibility criteria. The charting categories were refined iteratively during full-text assessment to maintain consistency in terminology across heterogeneous sources. Data charting was conducted by two investigators, and discrepancies were resolved through consultation with a third investigator when required. For each source, the extraction process identified the quality attributes discussed, any explicit composite metrics proposed, the target hardware platform, software architecture context, evaluation method, trade-off dimensions, the reported strengths and limitations, and the relationship between the study and hardware-software co-design. The charted data were synthesized narratively and organized into thematic categories: foundational quality attributes, quantitative composite metrics, qualitative trade-off frameworks, user-centric evaluation, practical deployment profiles, and methodological trends. The complete 166-study corpus-mapping matrix is provided as Supplementary Table S1, whereas the main text integrates representative and grouped evidence selected to illustrate the principal methodological categories and their contribution to the research questions.

2.7. Critical Appraisal and Synthesis

Because the purpose of this scoping review was to map the breadth and characteristics of the evidence rather than to produce a pooled effect estimate, a formal risk-of-bias assessment was not performed. Instead, each included source was examined for its relevance

to the review questions, clarity of metric definition, explicitness of hardware constraints, and usefulness for trade-off analysis. Results were synthesized through descriptive tables, conceptual grouping, and narrative comparison across methodologies, hardware platforms, and quality attributes.

2.8. Methodological Limitations

The present review has methodological limitations despite the use of two-investigator screening, eligibility assessment, and data charting with third-investigator discrepancy resolution. In particular, the manuscript does not report formal inter-rater agreement statistics or an externally audited verification procedure across all review stages. As a result, the synthesis may still be affected by residual selection and extraction bias and should be interpreted as a structured descriptive mapping of the reviewed corpus rather than as an exhaustive error-free inventory. Future updates of this review would benefit from formal agreement reporting, externally auditable decision logs, and continued multi-reviewer consensus protocols across all review stages.

3. Results

This section presents the synthesized findings of the scoping review. Within the reviewed corpus, the results indicate a progression in evaluation methodologies, moving from a foundational vocabulary of individual quality attributes to broader multi-attribute evaluation approaches. These approaches support trade-off analysis that is increasingly relevant to contemporary hardware-software co-design.

3.1. Characteristics of Sources of Evidence

The included corpus comprises 166 sources of evidence addressing software architecture evaluation under hardware constraints, with emphasis on embedded systems, edge computing, heterogeneous accelerators, immersive systems, architectural quality assessment, and multi-objective optimization. The principal characteristics charted for each source were publication focus, target hardware platform, software architecture context, evaluation method, trade-off dimensions, and the presence of explicit or implicit composite metrics. These characteristics are reflected in the thematic syntheses presented throughout this section. Given the size of the corpus, the main text discusses representative studies in depth, maps a broader portion of the corpus by analytical role, and refers readers to Supplementary Table S1 for the complete high-level corpus-mapping matrix.

For complete corpus-level traceability, the additional included sources represented in Supplementary Table S1 are cited here as part of the evidence base [2–4,6–9,12–96].

To make the synthesis transparent and traceable, the charted evidence was mapped by analytical role in Table 2. This corpus mapping distinguishes the sources used to establish the software-architecture and quality vocabulary, the studies that report explicit multi-attribute optimization mechanisms, the evidence that documents constrained or specialized deployment platforms, and the sources that support benchmark-guided, service-oriented, or user-centric evaluation. The table therefore provides an academic traceability structure linking groups of studies to the research questions, while the surrounding subsections discuss the representative mechanisms in greater detail. The complete high-level source-by-source corpus mapping for all 166 included studies is reported separately in Supplementary Table S1.

Table 2. Analytical mapping of the reviewed corpus by contribution to the synthesis.

Corpus Role	Representative Evidence	Use in the Synthesis
Architecture-quality and assessment vocabulary	Quality metrics, open/modular architecture, technical debt, and complexity studies [1,10,11,97–100].	Grounds RQ1 with architecture-relevant quality vocabulary.
Explicit quantitative optimization and accelerator co-design	Hardware-aware NAS, CNN/BNN/transformer co-design, FPGA design, accelerators, and portable pipelines [3,8,101–115].	Supports RQ2 with explicit multi-attribute optimization mechanisms.
Embedded, edge, IoT, and resource-constrained deployment contexts	Low-end IoT, ARM edge, Raspberry Pi-class nodes, batteryless devices, embedded memory, real-time edge ML, NoC mapping, CubeSat inference, and HW/SW safety/security cases [116–134].	Supports RQ1 and RQ3 by linking metrics to constrained hardware.
Networked, service-oriented, cloud/fog, and virtualized infrastructures	QoS/QoE SDN, smart-city WiFi, cloud-fog/5G architectures, service-function chains, serverless edge, NFV/O-RAN, VNF benchmarking, and data-center energy studies [135–147].	Supports RQ2 and RQ3 for service-level trade-off evaluation.
Benchmark-guided, monitoring, and performance-diagnostic evidence	Browser benchmarking, system performance anomaly detection, query-performance prediction, and diagnostic studies [148–150].	Links individual measurements to benchmark-guided evaluation.
User-centric, interactive, and domain-specific outcome integration	XR testbeds and simulation scheduling for user experience [4,151].	Distinguishes user-centric outcomes from strict quantitative metrics.

Note: The table maps groups of studies to their analytical role in the synthesis; detailed mechanisms are discussed in the following subsections.

3.2. Foundational Quality Attributes and Metrics

The literature identifies a wide array of individual metrics used to assess specific aspects of software architecture. These metrics form the building blocks for more complex evaluations.

The ISO/IEC 25010:2011 standard provides a widely accepted framework for organizing these metrics, categorizing them under high-level quality characteristics [99,152]. Key characteristics relevant to this review include maintainability, performance efficiency, reliability, and portability. A comprehensive review conducted by Silva et al. [99] provides a catalog of these foundational attributes and their associated metrics. A summary of the most pertinent examples is presented in Table 3.

Within the reviewed corpus, ISO/IEC 25010:2011 serves mainly as a foundational software-quality reference for organizing single-attribute metrics. The review also identified other composition frameworks, including explicit objective functions for hardware-aware optimization, Quality Attribute Scenarios (QAS) for structured qualitative trade-off reasoning, and Quality-of-Experience (QoE)-oriented outcome integration in interactive systems. Beyond the ISO/IEC 25010 [152] vocabulary, the reviewed studies operationalize deployment-specific attributes such as latency, energy or power consumption, area, memory footprint, throughput, resource utilization, end-to-end inference latency, queueing delay, availability, error-detection coverage, timing correctness, communication overhead, and QoS/QoE-oriented outcomes when evaluating ARM-based edge devices, Raspberry Pi-class nodes, FPGA/eFPGA platforms, low-end IoT frameworks, batteryless devices, heterogeneous intelligent devices, real-time task chains, service-function-chaining environments, fog architectures, and interactive service platforms [102,107,116–118,120–122,128,130–133,135,139,143,153–155].

Table 3. Foundational Quality Attributes and Associated Metrics

Quality Characteristic	Quality Attribute	Associated Metrics	Reference
Maintainability	Modifiability	Average Weighted Degree, Dependency Concentration	[99]
Maintainability	Testability	Coupling/Decoupling	[99]
Maintainability	Cohesion	Cohesion	[99]
Maintainability	Reusability	Q-Value, In/Out Degree	[99]
Performance Efficiency	Performance	Response Time, Throughput	[99]
Reliability	Availability	Availability, Mean Time Between Failures (MTBF)	[99]
Reliability	Stability	Instability, Distance	[99]
Portability	Adaptability	Coupling/Decoupling	[99]
Portability	Portability	Dependency	[99]

While this catalog provides the necessary vocabulary for discussing architectural quality, individual metrics are insufficient for modern system design. They measure attributes in isolation, failing to capture the complex and often conflicting relationships between them. Effective co-design for resource-constrained hardware demands evaluation methods that make the trade-offs between these attributes explicit. Across the reviewed corpus, these individual metrics are subsequently combined or balanced in several recurring ways. Quantitative studies use explicit combinations such as accuracy-latency-energy-area [8], accuracy-latency-energy [3], accuracy-latency-energy-peak power [107], accuracy-power-latency-memory [154], computing-performance-memory-power models [128], resource-performance constraints in real-time edge machine-learning pipelines [131], communication cost and model-quality objectives under device heterogeneity [132], and mapping or memory-placement objectives for embedded and NoC-based systems [129,130]. Other studies rely on benchmark-driven comparisons of waiting time, receiving time, processing time, resource utilization, throughput, queuing behavior, timing correctness, and deadline-sensitive latency across ARM-based edge devices, embedded operating-system testbeds, real-time task chains, and virtualized service chains [121,133,139,155], or on QoS/QoE-oriented synthesis that links user satisfaction to technical parameters in service-oriented and simulation-based systems [135,151]. Taken together with the deployment-oriented synthesis later in the manuscript, Table 3 provides the single-attribute portion of the systematic metric inventory requested for this review.

3.3. Composite Evaluation Approaches in HW/SW Co-Design

Within the reviewed corpus, a recurrent pattern is the use of broader composite evaluation approaches to manage architectural trade-offs. However, only a small subset of the included studies explicitly defines architectural-level composite metrics through formal aggregation functions [3,8,102,106,107,154]. Most of the corpus instead relies on implicit multi-attribute reasoning, structured qualitative trade-off assessment, user-centric composite outcomes, or benchmark-guided system-level modeling [5,109,117,118,121,122,128–133,135,136,139,143,146,153,155,156]. Taken together, these distinctions provide a working taxonomy for interpreting how multi-attribute architectural evaluation is currently reported in the reviewed literature. These approaches can therefore be categorized into three primary forms, each suited to different design contexts: quantitative methods for

automated design space exploration, qualitative methods for reasoning about complex trade-offs in legacy or large-scale systems, and user-centric methods for optimizing end-to-end experiential systems.

Figure 2 summarizes the conceptual relationships among the identified evaluation approaches and now highlights a broader portion of the reviewed corpus, including quantitative studies such as FLASH [8], HW-FlowQ [3], EdgeTran [107], Reg-TuneV2 [154], and mixed-precision FPGA co-design [106], the qualitative Monarch case [5], and user-centric QoE-oriented examples such as ILLIXR [4] and service-oriented SDN evaluation [135].

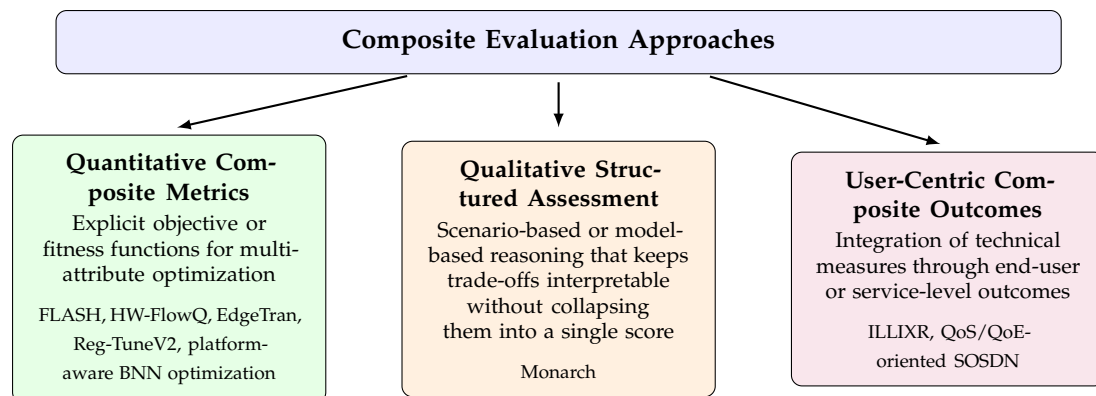


Figure 2. Conceptual map of composite evaluation approaches, illustrating the relationships among quantitative, qualitative, and user-centric forms of evaluation, together with representative methodologies reported in the reviewed corpus.

Together, Figure 2 and Table 4 provide the conceptual scaffold for interpreting the broader empirical synthesis presented in the following subsections.

Table 4. Comparison of composite evaluation approaches used for assessing software architectures and architecture-relevant software/hardware systems.

Approach	Core Characteristics	Strengths	Limitations	Representative Studies
Quantitative composite metrics	Optimization-based evaluation using objective functions and multi-objective formulations	Scalable, automatable, suitable for large design-space exploration	Highly sensitive to weight selection; potential bias in trade-off formulation	FLASH [8], HW-FlowQ [3], EdgeTran [107], Reg-TuneV2 [154], platform-aware BNN optimization [102]
Qualitative structured assessment	Scenario-based architectural reasoning using quality attribute scenarios and architectural views	High interpretability; effective for complex or legacy systems	Limited scalability; relies on expert judgment	Monarch (Google) [5]
User-centric composite outcomes	Evaluation driven by Quality-of-Experience (QoE) and perceptual metrics	Captures end-user impact directly; suitable for immersive systems	Difficult to replicate; influenced by contextual and human factors	ILLIXR [4], QoS/QoE-oriented SOSDN [135]

3.3.1. Quantitative Composite Metrics via Objective Functions

This approach uses a mathematical formula to combine multiple, distinct metrics into a single, optimizable score. This score serves as a direct, quantitative representation of the trade-offs between competing system goals, making it ideal for automated search and optimization algorithms.

Example 1 (FLASH Framework). *The FLASH framework for Neural Architecture Search (NAS), as reported in [8], exemplifies this approach with the objective function*

$$f_{obj} = \frac{\theta}{A \times L \times E}$$

where the expression is reproduced from the cited source rather than proposed in the present review. This function seeks to maximize test accuracy (θ) while simultaneously minimizing the hardware costs of chip area (A), inference Latency (L), and Energy consumption (E). By encapsulating these competing goals in a single equation, the framework can automatically explore the design space to find architectures that represent an optimal balance of software performance and hardware efficiency.

Example 2 (HW-FlowQ Framework). *Similarly, the HW-FlowQ co-design methodology uses a genetic algorithm guided by the fitness function $F_p = \left(1 - \frac{\Psi^* - \Psi}{t}\right) \cdot \log\left(\frac{\phi^*}{\phi}\right)$ [3]. This formula balances the goal of maintaining high accuracy (Ψ) against the benefits of improved hardware efficiency (ϕ , which can represent energy or latency). The logarithmic term rewards significant hardware gains, while the accuracy term penalizes solutions that deviate too far from a baseline, creating a powerful composite metric for navigating the HW-CNN co-design space.*

Beyond these two anchor cases, additional quantitative studies in the reviewed corpus combine or jointly optimize multiple deployment metrics rather than treating predictive accuracy alone as sufficient. Hardware platform-aware BNN optimization explicitly balances predictive performance against implementation cost on heterogeneous target hardware [102]. Reg-TuneV2 targets embedded deployment by jointly considering accuracy, power, latency, and memory-aware model-size constraints [154]. EdgeTran profiles transformer models across mobile edge platforms and co-optimizes accuracy, latency, energy consumption, and peak power draw [107]. Likewise, mixed-precision CPU co-design on FPGA minimizes energy and area under accuracy constraints [106], while bit-line computing for edge AI couples model compression and hardware decoding to improve energy efficiency with bounded accuracy degradation [108]. Hardware-specialized acceleration studies extend this pattern to resistive neural accelerators, FPGA-based similarity estimation, FPGA-supported SVM classification, and resource-efficient eFPGA-targeted classification, where accuracy, throughput, resource use, and energy or implementation cost are evaluated jointly [101,105,112,113]. Related optimization-oriented studies also treat resource-constrained scheduling and platform mapping as multi-attribute search problems, as illustrated by iterative compilation optimization guided by metric learning [157], exact modulo scheduling for resource-constrained CGRAs [156], embedded data-placement optimization [130], NoC-based real-time application mapping [129], and high-accuracy multi-FPGA acceleration strategies that balance throughput, parallelism, and hardware efficiency [109]. These studies broaden the quantitative portion of the reviewed corpus beyond the recurring FLASH/HW-FlowQ pair and show that the combined metrics vary according to deployment goals and hardware context.

3.3.2. Qualitative Trade-Off Assessment via Structured Analysis

This approach employs a formal, structured process for reasoning about architectural trade-offs without necessarily reducing them to a single numeric score. It is particularly effective for communicating the complex, multi-faceted impact of architectural changes in large-scale or legacy systems.

Case Study (Google Monarch): The architectural re-design of Google’s Monarch monitoring system illustrates this method [5]. Crucially, the team used Quality Attribute Scenarios (QAS) to translate vague goals such as “high availability” into specific, measurable targets (e.g., “ $\geq 99.99\%$ of queries return successfully”), which guided the entire analysis [5]. They then modeled the existing and proposed architectures using UML component and sequence diagrams. This structured modeling process enabled a clear, qualitative trade-off analysis, summarized in Table 5, which shows the anticipated impact of the new design. This model-based reasoning served as a form of qualitative multi-attribute assessment, making complex architectural trade-offs explicit and debatable without reducing them to a single numerical score.

Table 5. Qualitative trade-off analysis for Monarch re-design [5].

Quality Attribute	Impact of New Design
Availability	+ Improved
Maintainability	+ Improved
Latency	– Degraded
Resource efficiency	– Degraded

Note: The plus sign indicates an improvement, whereas the minus sign indicates a degradation relative to the previous design.

Monarch is the clearest example of structured qualitative assessment in the strict sense. However, the broader corpus also contains architecture-relevant trade-off methods that remain interpretable without collapsing all criteria into one score. For example, SOLAR uses multi-target design-space exploration to reason about performance, power, energy, and efficiency across CPU/GPU/FPGA execution paths [158], heterogeneous serverless edge orchestration tunes weights across throughput, response time, energy efficiency, AI accuracy, and cost to support deployment decisions [143], and queue-aware service-function evaluation plus load-aware in-server flow scheduling use bottleneck-sensitive latency metrics to guide deployment choices in softwarized infrastructures [139,146]. These studies broaden the notion of architectural trade-off methods even when they are not expressed as purely qualitative scenario tables.

3.3.3. User-Centric Evaluation via Composite Outcomes

This approach evaluates a system based on a high-level, holistic metric that directly captures the end-user’s perceived quality. This is especially relevant for interactive, user-facing systems where the interplay of many low-level system characteristics determines the final experience.

Case Study (ILLIXR Testbed): The Illinois Extended Reality (ILLIXR) project advocates for evaluating XR systems based on end-to-end Quality-of-Experience (QoE) [4]. QoE is a composite user-oriented outcome that integrates numerous underlying factors, including motion-to-photon latency, display refresh rate, rendering quality, and power consumption. A failure in any one of these dimensions can ruin the user’s sense of immersion. In this context, QoE drives architectural decisions because it directly measures the system’s ability to fulfill its primary purpose from the user’s perspective; in the present review, however,

it is treated as a user-centric evaluation approach rather than as a quantitative composite metric in the strict sense.

The reviewed corpus also contains user-centered integration strategies beyond XR. In service-oriented software-defined networking, QoS/QoE support is operationalized through adaptive prioritization, server selection, and multi-criteria routing based on an integral additive QoS criterion, thereby linking user satisfaction directly to technical network parameters [135]. Related service and simulation studies connect traffic control, scheduling, and response behavior to user-facing outcomes in smart-city WiFi and farm-simulation contexts [136,151]. This demonstrates that user-centric composite outcomes are not confined to immersive XR platforms, even though the underlying deployment settings differ substantially.

Beyond the conceptual comparison, Table 6 summarizes the practical deployment profiles identified in the reviewed corpus by relating software or system artifacts, deployment contexts, measured attributes, and composite evaluation mechanisms.

Table 6. Deployment-oriented synthesis of software or system artifacts evaluated on resource-constrained or specialized hardware.

Study	Software or System Artifact	Deployment Context	Measured Attributes	Trade-Off Integration Mechanism
FLASH [8]	CNN inference pipeline	In-memory computing accelerator	Accuracy, Latency, Energy, Area	Explicit objective function
HW-FlowQ [3]	CNN inference pipeline	FPGA/edge AI accelerator	Accuracy, Energy, Latency	Fitness function within genetic search
EdgeTran [107]	Transformer inference pipeline	Mobile edge devices	Accuracy, Latency, Energy, Peak Power	Device-aware co-search guided by hardware profiling
SOLAR [158]	Service-oriented deep learning architecture	CPU/GPU/FPGA heterogeneous platform	Performance, Power, Energy, Efficiency	Multi-target design-space exploration across heterogeneous accelerators
Arch2End [153]	Distributed embedded inference scheduler/model	Heterogeneous intelligent devices	End-to-End Inference Latency, Device Performance Features	Two-stage benchmark-guided system-level modeling
ChamelloT [117]	Hardware-assisted IoT OS framework	Low-end FPGA-assisted IoT device	API Latency, Thread Performance, FPGA Resource Consumption	Benchmark-based HW-assisted OS co-evaluation
Hedgi/Serverless edge [143]	Serverless edge application stack	Heterogeneous edge nodes	Throughput, Energy Efficiency, Response Time, AI Accuracy, Cost	Performance-driven empirical weight tuning and multi-objective balancing
ARM edge benchmarking [133]	Containerized edge software stack	ARM-based edge device	Waiting Time, Receiving Time, Processing Time, Resource Utilization, Throughput	Comparative benchmarking across container technologies
MultiRuler [128]	Embedded intelligent-task execution model	Jetson AGX Xavier/Cambricon MLU220	Computing Performance, Memory Usage, Power Consumption	Predictive multidimensional resource modeling
PHiLiP on the HiL [121]	Low-end embedded OS validation framework	Multi-platform hardware-in-the-loop testbed	Timing Behavior, Feature Correctness, Platform Compatibility	Comparative HiL-based multi-platform validation
Network Queuing Assessment [139]	Service-function chaining environment	Softwarized network service infrastructure	Queueing Delay, Throughput, CPU/Memory Bottleneck Indicators	Queue-aware bottleneck detection across chained services
Vina-FPGA-Cluster [109]	Molecular-docking acceleration pipeline	Multi-FPGA cluster	Accuracy, Throughput, Parallel Scalability, Energy Efficiency	Accuracy-preserving multi-level parallel hardware/software co-optimization
Model-Based 6-DoF Accelerator [159]	Physics-oriented model accelerator	FPGA-based computational platform	Numerical Error, Resource Utilization, Timing Performance	Model-based design with resource-sharing trade-off analysis
ILLIXR [4]	Immersive XR runtime stack	Interactive XR platform	Motion-to-photon Latency, Display Refresh Rate, Rendering Quality, Power Consumption	QoE-driven composite outcome

This table emphasizes deployment-oriented software or system artifacts rather than claiming that all examples are software architectures in the narrow formal sense.

3.4. Practical Deployment Profiles and Evaluation Targets

A practical question underlying this review is whether architecture-relevant software systems can in fact be deployed on resource-constrained hardware, or whether such

environments are limited to highly optimized low-level routines. Within the reviewed corpus, the answer is that architecture-relevant deployment is not restricted to simple procedural code alone. The evidence includes structured software or system artifacts such as convolutional neural network inference pipelines [3,8,102,108], transformer models on mobile edge platforms [107], hardware-assisted low-end IoT operating-system frameworks [117], containerized edge software stacks on ARM-based devices [133], heterogeneous intelligent-device schedulers and performance models [128,153], real-time edge machine-learning resource optimizers [131], service-oriented accelerator-backed deep learning systems [158], low-end embedded operating-system validation environments [121], energy-aware or batteryless IoT hardware contexts [120,122], fog and smart-city networked architectures [124,125,136], cryptographic or safety-oriented hardware/software co-design cases [116], multi-FPGA molecular-docking acceleration pipelines [109], model-based FPGA dynamical accelerators [159], and immersive XR runtime stacks [4] when these are paired with constrained or specialized hardware platforms. At the same time, programming language was not a charted variable in the present review, so direct comparison against simple C or C++ implementations is outside the scope of the available evidence.

The practical synthesis therefore focuses on software or system artifacts, deployment hardware, measured attributes, and the mechanism used to integrate trade-offs. Representative target platforms in the reviewed corpus span in-memory and FPGA-based accelerators [3,8,159], mobile and ARM-based edge devices [107,121,133], low-end reconfigurable IoT and batteryless platforms [117,122], Raspberry Pi-class computing nodes [120], heterogeneous intelligent-device environments such as Jetson AGX Xavier and Cambrian MLU220 [128,131], distributed embedded and NoC-based systems [129,153], multi-FPGA clusters [109], and specialized latency-sensitive XR platforms [4]. From the perspective of RQ3, the corpus does not converge on a single classical architecture pattern; instead, it recurrently addresses architecture-relevant artifact classes such as inference pipelines, service-oriented accelerator-backed systems, containerized edge application stacks, hardware-assisted OS frameworks, distributed embedded schedulers, service-function chains, fog-computing and smart-city service architectures, hardware/software co-designed cryptographic kernels, physics-oriented model accelerators, and XR runtime stacks [4,107,109,116,117,121,124,125,129,131,133,136,139,143,153,158,159]. Qualitative reference cases such as Monarch remain important conceptually, but because they target server-scale infrastructure rather than constrained deployment hardware, they are retained in Table 4 and not in the deployment-oriented synthesis below.

This deployment-oriented synthesis demonstrates that the reviewed literature is not limited to low-level hand-optimized routines; rather, it includes multiple classes of software or system artifacts integrated with constrained or specialized hardware and evaluated through explicit trade-off-aware mechanisms. It also broadens the narrative coverage of the reviewed corpus beyond the small recurring subset discussed in the previous version by incorporating low-end IoT frameworks, ARM-based edge benchmarking, heterogeneous intelligent-device modeling, serverless edge orchestration, hardware-in-the-loop OS validation, service-function bottleneck assessment, batteryless IoT platforms, embedded memory optimization, real-time edge ML resource optimization, and model-based FPGA acceleration. Together, Tables 3 and 6 provide a systematic inventory of the principal single-attribute metrics, deployment contexts, and composite evaluation mechanisms identified in the reviewed corpus.

3.5. Trends in Composite Evaluation Methodologies

The temporal analysis of the selected studies suggests an evolution in architectural evaluation practices within the reviewed sample. Earlier works relied more heavily on qual-

itative or single-metric evaluations, whereas later publications more frequently reported quantitative composite metrics and user-centric approaches.

Quantitative methodologies appear more frequently in later publications, particularly in automated design-space exploration, hardware-aware search, and resource-optimization frameworks such as FLASH [8], HW-FlowQ [3], EdgeTran [107], Reg-TuneV2 [154], mixed-precision FPGA co-design [106], resource-constrained CGRA mapping [156], real-time heterogeneous-edge ML optimization [131], and multi-objective federated learning under device heterogeneity [132]. In parallel, user-centric evaluation, especially Quality-of-Experience (QoE), becomes more visible in immersive and interactive systems and in service-oriented network settings, as exemplified by ILLIXR [4], QoS/QoE-oriented SOSDN evaluation [135], and simulation scheduling for user experience [151]. Figure 3 should therefore be interpreted as a descriptive synthesis of the reviewed corpus rather than as a direct field-wide measurement.

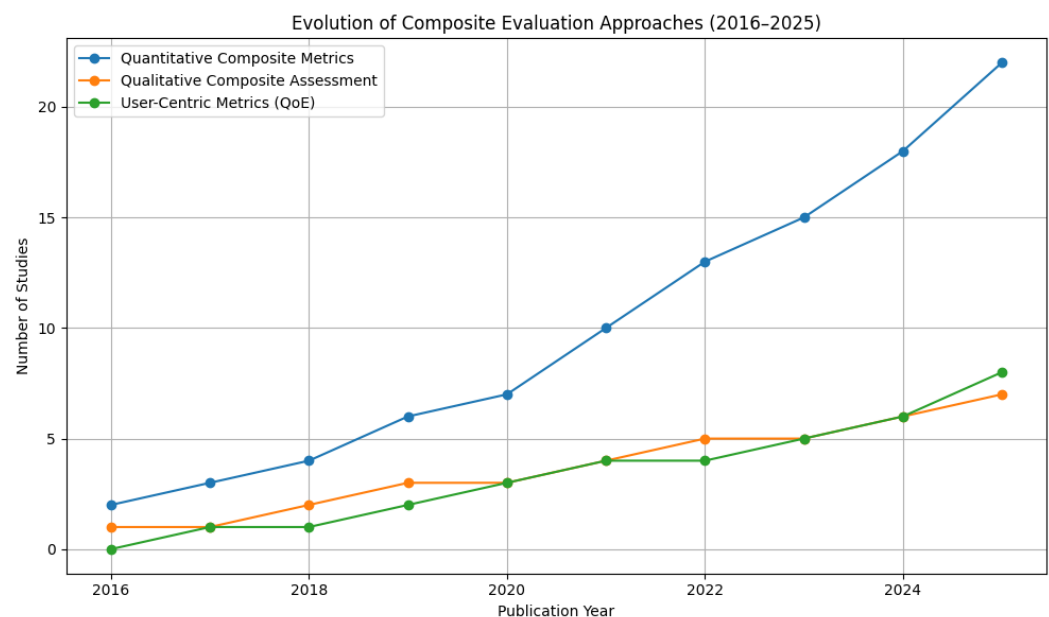


Figure 3. Descriptive trends in architectural evaluation approaches within the reviewed corpus (2016–2025), showing increased visibility of quantitative composite metrics and user-centric evaluation frameworks in later publications [3,4,8].

4. Discussion

4.1. Global Overview of Composite Evaluation Approaches

The results of this review carry significant strategic implications for the design and development of software on resource-constrained hardware. Within the reviewed corpus, the findings suggest a gradual broadening in how system quality is defined and measured, moving beyond isolated metrics toward more integrated perspectives on trade-offs. At the same time, the review also shows that explicit architectural-level composite metrics remain uncommon, and that much of the field still operates through implicit, benchmark-guided, or framework-based multi-attribute evaluation [102,109,117,118,121,128–133,135,139,143,153,156].

4.1.1. Key Trends

A prominent pattern in the reviewed sample is the movement away from isolated single-attribute optimization toward more holistic, multi-objective, and co-designed evaluation. For embedded and edge-oriented systems in particular, the reviewed studies suggest that software architecture is often evaluated in close relation to the hardware it

targets. Frameworks like FLASH [8], HW-FlowQ [3], EdgeTran [107], Reg-TuneV2 [154], and mixed-precision FPGA co-design [106] exemplify this concept by treating hardware and software parameters as interdependent variables in a unified optimization problem, while Arch2End [153], MultiRuler [128], ChamellIoT [117], and PHiLIP on the HiL [121] show the same tendency in system-level modeling and deployment-oriented embedded frameworks.

4.1.2. Challenges and Limitations

Despite their usefulness, composite evaluation approaches introduce their own set of challenges. A primary difficulty lies in maintaining operational consistency across evaluation forms that do not share the same level of formalization. Quantitative objective functions, qualitative trade-off frameworks, and user-centric composite outcomes cannot always be compared directly, even when they address related architectural decisions. In addition, selecting appropriate weighting coefficients for quantitative objective functions remains difficult. As noted in the context of software-defined networks, this is not a trivial task, especially in dynamic environments where priorities may shift [135]. An incorrect weighting can lead an automated search algorithm to a suboptimal solution. Furthermore, there is an inherent tension between conflicting goals that cannot always be resolved by a single formula. The Monarch case study [5] provides a clear example, where improving maintainability and availability came at the direct cost of increased latency and resource consumption. Qualitative frameworks are important in these scenarios because they facilitate human-led decision-making about which trade-offs are acceptable.

4.1.3. Impact on Development Practices

These evaluation methods have the potential to influence development practices substantially. The use of quantitative composite metrics within automated frameworks like NAS (FLASH [8]) and genetic algorithms (HW-FlowQ [3]) shifts part of architectural design from a purely manual, intuition-based process toward a more data-driven and partially automated one. This enables the exploration of larger design spaces than would normally be feasible through manual reasoning alone, thereby supporting earlier and more evidence-based architectural decisions.

4.1.4. Connection to Technical Debt

Ultimately, the adoption of composite evaluation approaches may serve as a useful mechanism for the proactive management of architectural technical debt [1]. This occurs when a team optimizes for a single, short-term metric (e.g., time-to-market or immediate performance) at the expense of other crucial attributes like maintainability, portability, or energy efficiency. While this may yield a short-term benefit, it can also accrue “interest” in the form of increased future development costs, higher operational expenses (e.g., battery drain), and reduced system resilience. A composite evaluation framework makes these longer-term costs more visible during design-time reasoning, thereby helping teams justify architectural decisions that prioritize long-term system health over myopic optimization.

4.2. Practical Implications for Software/Hardware Integration

From a practical perspective, the reviewed corpus indicates that deployment on resource-constrained hardware is not limited to simple low-level routines. Instead, the evidence base includes architecture-relevant software or system artifacts such as CNN and transformer inference pipelines [3,8,102,107,108], service-oriented accelerator-backed systems [158], low-end IoT operating-system frameworks [117], heterogeneous serverless edge stacks [143], ARM-based containerized edge deployments [133], distributed embedded scheduling environments [128,153], real-time heterogeneous-edge ML systems [131], low-end OS validation frameworks [121], service-function chains [139], fog and smart-

city service architectures [124,125,136], embedded memory and NoC-based application mapping cases [129,130], physics-oriented FPGA accelerators [159], multi-FPGA computational pipelines [109], and immersive XR runtime stacks [4]. These artifacts are evaluated through combinations of accuracy, latency, energy, area, resource utilization, throughput, memory usage, power, cost, communication overhead, queueing delay, timing correctness, and QoE-oriented outcomes, depending on the deployment context.

At the same time, the review does not support a direct language-level comparison between these artifacts and simple C or C++ implementations, because the programming language was not a charted variable. The practical implication of the present synthesis is therefore not a programming-language ranking, but a deployment-oriented mapping between software or system artifact classes, hardware constraints, and the metrics used to judge success at the software/hardware boundary.

5. Conclusions

The evaluation of architecture-relevant software or system artifacts for resource-constrained hardware has matured into a broader landscape of multi-attribute reasoning. Rather than treating all such methods as the same evaluative object, this review distinguishes explicit quantitative composite metrics from qualitative trade-off assessment and user-centric QoE-oriented evaluation. Within the reviewed corpus, these approaches collectively support hardware-software co-design by making architectural trade-offs more explicit, although explicit architectural-level composite metrics remain relatively rare. By synthesizing representative techniques from a broader portion of the reviewed corpus—including mathematical objective functions in automated search, benchmark-guided system-level modeling, deployment-oriented edge benchmarking, hardware-assisted IoT frameworks, service-oriented accelerator architectures, and QoE-oriented integration—this report offers a consolidated resource for researchers and practitioners navigating domain-specific system design under hardware constraints.

Future Work

The source documents highlight several promising avenues for future research that build directly upon the findings discussed in this report. In particular, the review suggests the need for a more operational and architecture-centered research agenda. Key directions include:

1. **Operational definitions of architectural-level composite metrics:** Future work should establish clearer formal criteria for distinguishing explicit composite metrics from broader multi-attribute evaluation approaches, especially in studies that combine software and hardware concerns without providing an explicit aggregation function.
2. **Mapping evaluation approaches to architectural decisions:** The reviewed corpus would benefit from follow-up studies that connect composite evaluation approaches more directly to software architecture patterns, styles, tactics, and design decisions, thereby improving the practical interpretability of the evaluation results.
3. **Reproducible multi-reviewer evidence synthesis:** Updated reviews should incorporate dual independent screening, duplicate charting, and consensus-based conflict resolution in order to improve reproducibility and reduce selection and extraction bias.
4. **Cross-domain validation of user-centric and hardware-aware approaches:** Building on examples such as ILLIXR and hardware-aware optimization frameworks, future research should examine whether user-centric and optimization-driven evaluation approaches can be translated into reusable architecture-level assessment procedures across multiple application domains.

Supplementary Materials: The following supporting information can be downloaded at: <https://www.mdpi.com/article/10.3390/app16136436/s1>, Table S1: Complete Corpus Mapping Matrix (166 Studies).

Author Contributions: Conceptualization, A.I.L.C. and J.R.G.R.; methodology, A.I.L.C.; investigation, A.I.L.C., J.R.G.R., C.E.B.y.G.L., A.J.L.V., R.S.A. and V.I.R.A.; formal analysis, A.I.L.C.; writing—original draft preparation, A.I.L.C.; writing—review and editing, A.I.L.C., J.R.G.R., C.E.B.y.G.L., A.J.L.V., R.S.A. and V.I.R.A.; visualization, A.I.L.C.; supervision, J.R.G.R.; project administration, A.I.L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. Funding sources of the included evidence were not charted consistently across the analyzed corpus and therefore were not synthesized separately in this review.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The protocol for this scoping review is openly available in OSF Registries (DOI: <https://doi.org/10.17605/OSF.IO/U85BV>; URL: <https://osf.io/u85bv/> (accessed on 14 June 2026)). The high-level corpus mapping for the 166 included sources is provided in Supplementary Table S1. The data analyzed in this review are derived from the publications cited in the reference list.

Acknowledgments: The authors thank their academic institutions for their support during the preparation of this manuscript. During the preparation of this manuscript, the authors used OpenAI ChatGPT (GPT-5 family) for proofreading, formatting support, and assistance in organizing the supplementary corpus-mapping matrix. The authors reviewed and edited all outputs and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

HW/SW	Hardware-software
IoT	Internet of Things
NAS	Neural Architecture Search
PRISMA-ScR	Preferred Reporting Items for Systematic Reviews and Meta-Analyses extension for Scoping Reviews
QoE	Quality of Experience
QAS	Quality Attribute Scenarios
XR	Extended Reality

References

1. Junior, A.A.C.; Misra, S.; Soares, M.S. A Maturity Model based on ISO/IEC/IEEE 42010:2011 to Identify Technical Debt in Software Architecture. In Proceedings of the International Conference on Computational Science and Its Applications (ICCSA), Saint Petersburg, Russia, 1–4 July 2019; pp. 31–42.
2. Kiaei, P.; Yao, Y.; Liu, Z.-Y.; Fern, N.; Breunese, C.-B.; Van Woudenberg, J.; Gillis, K.; Dich, A.; Grossmann, P.; Schaumont, P. Gate-Level Side-Channel Leakage Ranking with Architecture Correlation Analysis. *IEEE Trans. Emerg. Top. Comput.* **2024**, *12*, 496–507. [[CrossRef](#)]
3. Fasfous, N.; Vemparala, M.R.; Frickenstein, A.; Valpreda, E.; Salihu, D.; Doan, N.A.V.; Unger, C.; Nagaraja, N.S.; Martina, M.; Stechele, W. HW-FlowQ: A Multi-Abstraction Level HW-CNN Co-design Quantization Methodology. *ACM Trans. Embed. Comput. Syst.* **2021**, *20*, 66. [[CrossRef](#)]
4. Huzaifa, M.; Desai, R.; Grayson, S.; Jiang, X.T.; Jing, Y.; Lee, J.; Lu, F.; Pang, Y.H.; Ravichandran, J.; Sinclair, F.; et al. ILLIXR: An Open Testbed to Enable Extended Reality Systems Research. *IEEE Micro* **2022**, *42*, 97–106. [[CrossRef](#)]
5. Jia, Q.; Cai, Y.; Çakmak, O.C. A Model-based, Quality Attribute-guided Architecture Re-Design Process at Google. In Proceedings of the 45th International Conference on Software Engineering: SEIP, Melbourne, Australia, 17–19 May 2023.

6. Wang, R.X.; Moon, S.H.; Hu, X.S.; Jiao, X.; Reis, D. A Computing-in-Memory-Based One-Class Hyperdimensional Computing Model for Outlier Detection. *IEEE Trans. Comput.* **2024**, *73*, 1559–1574. [\[CrossRef\]](#)
7. Choi, H.-J.; Um, J.-Y. Design and HDL Implementation of Pulse-Arrival-Time Estimation Using XGBoost Regression with Tree-Recycling Architecture. *IEEE Access* **2025**, *13*, 27596–27609. [\[CrossRef\]](#)
8. Li, G.H.; Mandal, S.K.; Ogras, U.Y.; Marculescu, R. FLASH: Fast Neural Architecture Search with Hardware Optimization. *ACM Trans. Embed. Comput. Syst.* **2021**, *20*, 63.
9. Xin, Y.; Li, W.J.; Jia, C.J.; Li, X.F.; Xu, Y.; Liu, B.; Tian, Z.H.; Zhang, W.Z. Recursive Multi-Tree Construction with Efficient Rule Sifting for Packet Classification on FPGA. *IEEE/ACM Trans. Netw.* **2024**, *32*, 1707–1722. [\[CrossRef\]](#)
10. Guertin, N.; Schmidt, D. A Modular Open Systems Approach (MOSA) Maturity Model. In Proceedings of the 21st Annual Acquisition Research Symposium, Monterey, CA, USA, 8–9 May 2024.
11. Open Architecture Assessment Team (OAET). *Naval Open Systems Architecture Strategy, Open Architecture Assessment Tool*; OAET: Patuxent River, MD, USA, 2014.
12. Guertin, N.; Miller, R. A-RCI—The Right Way to Submarine Superiority. *Nav. Eng. J.* **1998**, *110*, 21–33. [\[CrossRef\]](#)
13. Open Systems Joint Task Force (OSJTF). *MOSA Program Assessment and Rating Tool*; OSJTF: Arlington, VA, USA, 2004.
14. Cassano, L.; Iamundo, M.; Lopez, T.A.; Nazzari, A.; Di Natale, G. DETON: Defeating hardware Trojan horses in microprocessors through software Obfuscation. *J. Syst. Archit.* **2022**, *129*, 102592. [\[CrossRef\]](#)
15. Rajendran, S.R.; Dipu, N.F.; Tarek, S.; Kamali, H.M.; Farahmandi, F.; Tehranipoor, M. Exploring the Abyss? Unveiling Systems-on-Chip Hardware Vulnerabilities Beneath Software. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 3914–3926. [\[CrossRef\]](#)
16. Ryu, D.; Kim, Y.; Hur, J. γ -Knife: Extracting Neural Network Architecture Through Software-Based Power Side-Channel. *IEEE Trans. Dependable Secur. Comput.* **2024**, *21*, 2686–2703. [\[CrossRef\]](#)
17. Bhaskar, C.V.; Munaswamy, P. An Efficient Reconfigurable Architecture for Software Defined Radio. *IEICE Trans. Inf. Syst.* **2023**, *106*, 1519–1527. [\[CrossRef\]](#)
18. Hong, Y.; Ning, Z.H.; Sun, Y.; Shen, C.X. A dual-system trusted computing node construction method based on ARM multi-core CPU architecture. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e7037. [\[CrossRef\]](#)
19. Gruber, F.; Bandle, M.; Engelke, A.; Neumann, T.; Giceva, J. Bringing Compiling Databases to RISC Architectures. *Proc. VLDB Endow.* **2023**, *16*, 1222–1234. [\[CrossRef\]](#)
20. Sini, J.; Scialabba, K.; Violante, M.; Cosimi, F.; Arena, A. A Novel Approach to Address Random Hardware Failures for Automotive Application Within the ISO26262 and AUTOSAR Frameworks. *IEEE Access* **2024**, *12*, 165845–165860. [\[CrossRef\]](#)
21. Nikulchev, E.; Ilin, D.; Kolyasnikov, P.; Magomedov, S.; Alexeenko, A.; Kosenkov, A.N.; Sokolov, A.; Malykh, A.; Ismatullina, V.; Malykh, S. Isolated Sandbox Environment Architecture for Running Cognitive Psychological Experiments in Web Platforms. *Future Internet* **2021**, *13*, 245. [\[CrossRef\]](#)
22. Li, L.L.; Liu, J.Y.; Fan, J.P.; Qian, X.H.; Hwang, K.; Chung, Y.C.; Yu, Z.B. SOCA-DOM: A Mobile System-on-Chip Array System for Analyzing Big Data on the Move. *J. Comput. Sci. Technol.* **2022**, *37*, 1271–1289. [\[CrossRef\]](#)
23. Javeed, K.; Gregg, D. Efficient Number Theoretic Transform Architecture for CRYSTALS-Kyber. *IEEE Trans. Circuits Syst. II Express Briefs* **2025**, *72*, 263–267. [\[CrossRef\]](#)
24. Yacoub, M.H.; Ismail, S.M.; Said, L.A.; Madian, A.H.; Radwan, A.G. Reconfigurable hardware implementation of K-nearest neighbor algorithm on FPGA. *AEU-Int. J. Electron. Commun.* **2024**, *173*, 154999. [\[CrossRef\]](#)
25. Chen, X.; Zhao, Y.T.; Guo, A.; Chen, J.W.; Dong, F.Y.; Zhang, Z.Y.; Xiong, T.Z.; Wang, B.; Kong, Y.Y.; Si, X. Toggle Rate Aware Quantization Model Based on Digital Floating-Point Computing-In-Memory Architecture. *IEEE Trans. Circuits Syst. II Express Briefs* **2024**, *71*, 3181–3185. [\[CrossRef\]](#)
26. Xu, H.; Bai, L.T.; Huang, W. An optimization-inspired intrusion detection model for software-defined networking. *Electron. Res. Arch.* **2025**, *33*, 231–254. [\[CrossRef\]](#)
27. Lonbar, S.M.; Beigi, A.; Bagheri, N.; Peris-Lopez, P.; Camara, C. Deep learning based bio-metric authentication system using a high temporal/frequency resolution transform. *Front. Digit. Health* **2024**, *6*, 1463713. [\[CrossRef\]](#)
28. Freire, P.; Srivallapanondh, S.; Spinnler, B.; Napoli, A.; Costa, N.; Prilepsky, J.E.; Turitsyn, S.K. Computational Complexity Optimization of Neural Network-Based Equalizers in Digital Signal Processing: A Comprehensive Approach. *J. Light. Technol.* **2024**, *42*, 4177–4201. [\[CrossRef\]](#)
29. Zografopoulos, I.; Ospina, J.; Liu, X.R.; Konstantinou, C. Cyber-Physical Energy Systems Security: Threat Modeling, Risk Assessment, Resources, Metrics, and Case Studies. *IEEE Access* **2021**, *9*, 29775–29818. [\[CrossRef\]](#)
30. Li, Z.D.; Shang, C.Q.; Wu, J.J.; Li, Y. Microservice extraction based on knowledge graph from monolithic applications. *Inf. Softw. Technol.* **2022**, *150*, 106992. [\[CrossRef\]](#)
31. Nithyoosha, T.; Christopher, P.R. A survey on motion estimation and de-hazing algorithms and architectures. *Digit. Signal Process.* **2023**, *140*, 104130. [\[CrossRef\]](#)

32. Chatterjee, R.; Paluh, J.L.L.; Chowdhury, S.; Mondal, S.; Raha, A.; Mukherjee, A. SyNC, a Computationally Extensive and Realistic Neural Net to Identify Relative Impacts of Synaptopathy Mechanisms on Glutamatergic Neurons and Their Networks in Autism and Complex Neurological Disorders. *Front. Cell. Neurosci.* **2021**, *15*, 674030. [[CrossRef](#)] [[PubMed](#)]
33. Arney, D.; Zhang, Y.; Kennedy-Metz, L.R.; Dias, R.D.; Goldman, J.M.; Zenati, M.A. An Open-Source, Interoperable Architecture for Generating Real-Time Surgical Team Cognitive Alerts from Heart-Rate Variability Monitoring. *Sensors* **2023**, *23*, 3890. [[CrossRef](#)] [[PubMed](#)]
34. Brohet, M.; Regazzoni, F. A Survey on Thwarting Memory Corruption in RISC-V. *ACM Comput. Surv.* **2024**, *56*, 28.
35. Babun, L.; Aksu, H.; Uluagac, A.S. CPS Device-Class Identification via Behavioral Fingerprinting: From Theory to Practice. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 2413–2428. [[CrossRef](#)]
36. Wu, J.F.; Xu, M.X.; He, Y.Y.; Ye, K.J.; Xu, C.Z. Cloudnativesim: A Toolkit for Modeling and Simulation of Cloud-Native Applications. *Softw. Pract. Exp.* **2025**, *55*, 1185–1208. [[CrossRef](#)]
37. Pudi, D.; Yang, Y.; Stathis, D.; Prajapati, S.K.; Boppu, S.; Hemani, A.; Cenkeramaddi, L.R. Application Level Synthesis: Creating Matrix-Matrix Multiplication Library: A Case Study. *IEEE Access* **2024**, *12*, 155885–155903. [[CrossRef](#)]
38. Rafiev, A.; Yakovlev, A.; Tarawneh, G.; Naylor, M.F.; Moore, S.W.; Thomas, D.B.; Bragg, G.M.; Vousden, M.L.; Brown, A.D. Synchronization in graph analysis algorithms on the Partially Ordered Event-Triggered Systems many-core architecture. *IET Comput. Digit. Tech.* **2022**, *16*, 71–88. [[CrossRef](#)]
39. Uzuner, H.; Kavun, E.B. NLU-V: A Family of Instruction Set Extensions for Efficient Symmetric Cryptography on RISC-V. *Cryptography* **2024**, *8*, 9. [[CrossRef](#)]
40. Goodwill, J.M.; Prasad, N.; Hoskins, B.D.; Daniels, M.W.; Madhavan, A.; Wan, L.; Santos, T.S.; Tran, M.; Katine, J.A.; Braganca, P.M.; et al. Implementation of a Binary Neural Network on a Passive Array of Magnetic Tunnel Junctions. *Phys. Rev. Appl.* **2022**, *18*, 14039. [[CrossRef](#)]
41. Djomo, A.F.; Tiedeu, A.; Fotsing, J. Design and implementation in an Altera's cyclone IV EP4CE6E22C8 FPGA board of a fast and robust cipher using combined 1D maps. *IET Image Process.* **2024**, *18*, 1823–1843. [[CrossRef](#)]
42. Jurgielewicz, P.; Fiutowski, T.; Kublik, E.; Skoczen, A.; Szypulska, M.; Wiacek, P.; Hottowy, P.; Mindur, B. Modular Data Acquisition System for Recording Activity and Electrical Stimulation of Brain Tissue Using Dedicated Electronics. *Sensors* **2021**, *21*, 4423. [[CrossRef](#)] [[PubMed](#)]
43. Iwendi, C.; Rehman, S.U.; Javed, A.R.; Khan, S.; Srivastava, G. Sustainable Security for the Internet of Things Using Artificial Intelligence Architectures. *ACM Trans. Internet Technol.* **2021**, *21*, 73. [[CrossRef](#)]
44. Guler, E.; Karakus, M.; Ayaz, F. Genetic algorithm enabled virtual multicast tree embedding in Software-Defined Networks. *J. Netw. Comput. Appl.* **2023**, *209*, 103538. [[CrossRef](#)]
45. Bandara, E.; Shetty, S.; Tosh, D.; Liang, X.P. Vind: A Blockchain-Enabled Supply Chain Provenance Framework for Energy Delivery Systems. *Front. Blockchain* **2021**, *4*, 607320. [[CrossRef](#)]
46. Liu, C.X.; Sabu, A.; Chaudhari, A.; Kang, Q.X.; Carlson, T.E. Pac-Sim: Simulation of Multi-threaded Workloads using Intelligent, Live Sampling. *ACM Trans. Archit. Code Optim.* **2024**, *21*, 81. [[CrossRef](#)]
47. Murali, P.; Debroy, D.M.; Brown, K.R.; Martonosi, M. Toward Systematic Architectural Design of Near-Term Trapped Ion Quantum Computers. *Commun. ACM* **2022**, *65*, 101–109. [[CrossRef](#)]
48. Qin, T.; Li, C.; Liu, G.F. Control infrastructure monitoring system at the NSRL facility cluster. *J. Instrum.* **2022**, *17*, P11005. [[CrossRef](#)]
49. Li, H.Y.; Lu, H.; Li, X.W. Mortar-FP8: Morphing the Existing FP32 Infrastructure for High-Performance Deep Learning Acceleration. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2024**, *43*, 878–891. [[CrossRef](#)]
50. Charitopoulos, G.; Papaefstathiou, I.; Pnevmatikatos, D.N. Creating Customized CGRAs for Scientific Applications. *Electronics* **2021**, *10*, 445. [[CrossRef](#)]
51. Das Robin, D.; Khan, J.I. An open-source P416 compiler backend for reconfigurable match-action table switches: Making networking innovation accessible. *Comput. Netw.* **2024**, *242*, 110246. [[CrossRef](#)]
52. Baazaoui, M.K.; Ketata, I.; Fakhfakh, A.; Derbel, F. Modeling of Packet Error Rate Distribution Based on Received Signal Strength Indications in OMNeT plus plus for Wake-Up Receivers. *Sensors* **2023**, *23*, 2394. [[CrossRef](#)] [[PubMed](#)]
53. Fanti, A.; Perez, C.C.; Denis-Courmont, R.; Roascio, G.; Ekberg, J.E. Toward Register Spilling Security Using LLVM and ARM Pointer Authentication. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2022**, *41*, 3757–3766. [[CrossRef](#)]
54. Ignatius, T.M.; Singha, T.B.; Palathinkal, R.P. Power Side-Channel Attacks on Crypto-Core Based on RISC-V ISA for High-Security Applications. *IEEE Access* **2024**, *12*, 150230–150248. [[CrossRef](#)]
55. P., S.; Kavin, B.P.; Srividhya, S.R.; V., R.; C., K.; Lai, W.-C. Performance Evaluation of Stateful Firewall-Enabled SDN with Flow-Based Scheduling for Distributed Controllers. *Electronics* **2022**, *11*, 3000.
56. Lopez, A.B.; Jin, W.L.; Al Faruque, M.A. Attack Modeling Methodology and Taxonomy for Intelligent Transportation Systems. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 13255–13264. [[CrossRef](#)]

57. Karim, M.; Khalid, S.; Aleryani, A.; Khan, J.; Ullah, I.; Ali, Z. Human Action Recognition Systems: A Review of the Trends and State-of-the-Art. *IEEE Access* **2024**, *12*, 36372–36390. [\[CrossRef\]](#)
58. Dangl, T.; Sentanoe, S.; Reiser, H.P. Active and passive virtual machine introspection on AMD and ARM processors. *J. Syst. Archit.* **2024**, *149*, 103101. [\[CrossRef\]](#)
59. Baumann, C.; Martinoli, A. A modular functional framework for the design and evaluation of multi-robot navigation. *Robot. Auton. Syst.* **2021**, *144*, 103849. [\[CrossRef\]](#)
60. Dekhovich, A.; Tax, D.M.J.; Sluiter, M.H.F.; Bessa, M.A. Neural network relief: A pruning algorithm based on neural activity. *Mach. Learn.* **2024**, *113*, 2597–2618. [\[CrossRef\]](#)
61. Amponis, G.; Lagkas, T.; Sarigiannidis, P.; Vitsas, V.; Fouliras, P. Inter-UAV Routing Scheme Testbeds. *Drones* **2021**, *5*, 2.
62. Sgherzi, F.; Siracusa, M.; Fernandez, I.; Armejach, A.; Moretó, M. SpChar: Characterizing the sparse puzzle via decision trees. *J. Parallel Distrib. Comput.* **2024**, *192*, 104941. [\[CrossRef\]](#)
63. Charitopoulos, G.; Pnevmatikatos, D.N.; Gaydadjiev, G. MC-DeF: Creating Customized CGRAs for Dataflow Applications. *ACM Trans. Archit. Code Optim.* **2021**, *18*, 26. [\[CrossRef\]](#)
64. Challita, F.; Liénard, M.; Gaillot, D.P.; Molina-Garcia-Pardo, J.M. Evaluation of an Antenna Selection Strategy for Reduced Massive MIMO Complexity. *Radio Sci.* **2021**, *56*, e2020RS007242. [\[CrossRef\]](#)
65. da Silva, C.A.C.F.; Rosa, D.C.; Miranda, P.B.C.; Si, T.P.; Cerri, R.; Basgalupp, M.P. Automated CNN optimization using multi-objective grammatical evolution. *Appl. Soft Comput.* **2024**, *151*, 111124. [\[CrossRef\]](#)
66. Caforio, A.; Banik, S.; Todo, Y.; Meier, W.; Isobe, T.; Liu, F.K.; Zhang, B. Perfect Trees: Designing Energy-Optimal Symmetric Encryption Primitives. *IACR Trans. Symmetric Cryptol.* **2021**, *2021*, 36–73. [\[CrossRef\]](#)
67. Rojas-Muñoz, L.F.; Sánchez-Solano, S.; Martínez-Rodríguez, M.C.; Brox, P. On-Line Evaluation and Monitoring of Security Features of an RO-Based PUF/TRNG for IoT Devices. *Sensors* **2023**, *23*, 4070. [\[CrossRef\]](#) [\[PubMed\]](#)
68. de Lima, E.C.; Rossi, F.D.; Luizelli, M.C.; Calheiros, R.N.; Lorenzon, A.F. A neural network framework for optimizing parallel computing in cloud servers. *J. Syst. Archit.* **2024**, *150*, 103131. [\[CrossRef\]](#)
69. Alabdulatif, A.; Khalil, I.; Rahman, M.S. Security of Blockchain and AI-Empowered Smart Healthcare: Application-Based Analysis. *Appl. Sci.* **2022**, *12*, 11039. [\[CrossRef\]](#)
70. Toreini, E.; Mehrnezhad, M.; van Moorsel, A. Fairness as a Service (FaaS): Verifiable and privacy-preserving fairness auditing of machine learning systems. *Int. J. Inf. Secur.* **2024**, *23*, 981–997.
71. Tang, L.; Wang, K.; Zhang, Y.; Zhou, X.L.; Chen, Q.B. Service Function Chain Anomaly Detection Based on Distributed Generative Adversarial Network in Network Slicing Scenario. *J. Electron. Inf. Technol.* **2023**, *45*, 262–271.
72. Muro, F.; Baena, E.; Fortes, S.; Nielsen, L.; Barco, R. Noisy Neighbour Impact Assessment and Prevention in Virtualized Mobile Networks. *IEEE Trans. Netw. Serv. Manag.* **2023**, *20*, 415–425. [\[CrossRef\]](#)
73. Ramasubramanian, A.K.; Kazasidis, M.; Fay, B.; Papakostas, N.; Doulamis, A. On the Evaluation of Diverse Vision Systems towards Detecting Human Pose in Collaborative Robot Applications. *Sensors* **2024**, *24*, 578. [\[CrossRef\]](#) [\[PubMed\]](#)
74. Zafar, S.; Ayub, U.; Alkhamash, H.I.; Ullah, N. Framework for Efficient Auto-Scaling of Virtual Network Functions in a Cloud Environment. *Sensors* **2022**, *22*, 7597. [\[CrossRef\]](#) [\[PubMed\]](#)
75. Dipu, M.T.A.; Hossain, S.S.; Arafat, Y.; Rafiq, F.B. Real-time Driver Drowsiness Detection using Deep Learning. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*, 844–850. [\[CrossRef\]](#)
76. da Silva, M.A.; Garcia, R.D.; Carlo, J.C. Multi-objective optimization algorithms for building performance assessment—A benchmark. *Int. J. Archit. Comput.* **2024**, *24*, 35–59. [\[CrossRef\]](#)
77. Lee, W.; Lee, J.; Park, B.K.; Kim, R.Y.C. Microarchitectural Characterization on a Mobile Workload. *Appl. Sci.* **2021**, *11*, 1225. [\[CrossRef\]](#)
78. Chowdhury, R.; Talhi, C.; Ould-Slimane, H.; Mourad, A. Proactive and Intelligent Monitoring and Orchestration of Cloud-Native IP Multimedia Subsystem. *IEEE Open J. Commun. Soc.* **2024**, *5*, 139–155. [\[CrossRef\]](#)
79. Borylo, P.; Davoli, G.; Rzepka, M.; Lason, A.; Cerroni, W. Unified and standalone monitoring module for NFV/SDN infrastructures. *J. Netw. Comput. Appl.* **2021**, *175*, 102934. [\[CrossRef\]](#)
80. Jha, A.V.; Appasani, B.; Ghazali, A.; Bizon, N. A Comprehensive Risk Assessment Framework for Synchronphasor Communication Networks in a Smart Grid Cyber Physical System with a Case Study. *Energies* **2021**, *14*, 3428. [\[CrossRef\]](#)
81. Cheng, J.W.; Jiang, X.Y.; Zhou, H.L.; Dong, J.J. Advances and Challenges of Optoelectronic Intelligent Computing. *Chin. J. Lasers* **2022**, *49*, 1219001.
82. Mukhopadhyay, N.; Tewari, B.P. Cost and energy aware migration through dependency analysis of VM components in virtual cloud infrastructure. *Computing* **2025**, *107*, 24.
83. Napoli, O.; Duarte, D.; Alves, P.; Soto, D.H.P.; de Oliveira, H.E.; Rocha, A.; Boccato, L.; Borin, E. A benchmark for domain adaptation and generalization in smartphone-based human activity recognition. *Sci. Data* **2024**, *11*, 1192. [\[CrossRef\]](#) [\[PubMed\]](#)
84. AlShboul, R.; Khasawneh, A.M.; Al-Rousan, R.; Kumar, A.; Kumar, S.; Kaiwartya, O. D3S-Drone Security Surveillance System for Connected Drone Environment. *Adhoc Sens. Wirel. Netw.* **2023**, *57*, 209–225.

85. Parks, Z.; Carignan-Dugas, A.; Gustafson, E.; Meurice, Y.; Dreher, P. Applying the noiseless extrapolation error mitigation protocol to calculate real-time quantum field theory scattering phase shifts. *Phys. Rev. D* **2024**, *109*, 14505. [\[CrossRef\]](#)
86. Agrawal, N. Autonomic cloud computing based management and security solutions: State-of-the-art, challenges, and opportunities. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4349. [\[CrossRef\]](#)
87. Guo, P.F.; Yan, Y.J.; Zhang, F.; Zhu, C.S.; Zhang, L.C.; Dai, Z.B. Extending the classical side-channel analysis framework to access-driven cache attacks. *Comput. Secur.* **2023**, *129*, 103255. [\[CrossRef\]](#)
88. Lin, M.M.; Zhang, L.W.; Wang, Z.B.; Liu, H.Y.; Wang, K.Q.; Tang, G.Z.; Wang, W.K.; Sun, P. A combined system with convolutional neural networks and transformers for automated quantification of left ventricular ejection fraction from 2D echocardiographic images. *Intell. Med.* **2025**, *5*, 46–53. [\[CrossRef\]](#)
89. Benavides-López, J.A.; Rodríguez-Bulnes, J.; Collazo, J.R. Workflow for high definition documentation of the roman archaeological site of Herrera (Seville). *Disegnarecon* **2023**, *16*, 2.1–2.10.
90. Garcia, L.; Vazquez, E.; Sanchez, G.; Avalos, J.G.; Sanchez, G. An ultra-compact and high-speed FFT-based large-integer multiplier for fully homomorphic encryption using a dual spike-based arithmetic circuit over GF(p). *Neurocomputing* **2022**, *507*, 54–66. [\[CrossRef\]](#)
91. Martínez-Ruedas, C.; Adame-Rodríguez, F.J.; Díaz-Cabrera, J.M. Integrating and interconnecting of older SINUMERIK CNC machines with industry 4.0 using a plug-and-play system. *J. Ind. Inf. Integr.* **2024**, *38*, 100583. [\[CrossRef\]](#)
92. Costanzo, M.; Rucci, E.; García-Sánchez, C.; Naiouf, M. Viability Study of SYCL as a Unified Programming Model for Heterogeneous Systems Based on GPUs in Bioinformatics. *J. Comput. Sci. Technol.* **2024**, *24*, 193–194. [\[CrossRef\]](#)
93. Perrelli, M.; Cosco, F.; Carbone, G.; Lenzo, B.; Mundo, D. On the Benefits of Using Object-Oriented Programming for the Objective Evaluation of Vehicle Dynamic Performance in Concurrent Simulations. *Machines* **2021**, *9*, 41. [\[CrossRef\]](#)
94. Deshpande, P.; Kothari, A.G. Triple DES with radial basis function networks for securing in process of CCTV footage images. *Comput. Inform.* **2024**, *43*, 633–666. [\[CrossRef\]](#)
95. Nizamani, A.H.; Chen, Z.G.; Nizamani, A.A.; Bhatti, U.A. Advance brain tumor segmentation using feature fusion methods with deep U-Net model with CNN for MRI data. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 101793. [\[CrossRef\]](#)
96. Beskopylny, A.N.; Stel'makh, S.A.; Shcherban', E.M.; Razveeva, I.; Kozhakin, A.; Meskhi, B.; Chernil'nik, A.; Elshaeva, D.; Ananova, O.; Giry, M.; et al. Computer Vision Method for Automatic Detection of Microstructure Defects of Concrete. *Sensors* **2024**, *24*, 4373. [\[CrossRef\]](#) [\[PubMed\]](#)
97. Gagliardi, M.; Konrad, M.; Schmidt, D. *Addressing Open Architecture in Software Cost Estimation*; SEI Blog; Carnegie Mellon University: Pittsburgh, PA, USA, 2020.
98. Office of the Under Secretary of Defense for Research and Evaluation (OUSD(R&E)). *DoD and Military Services MOSA Instructions and Guidance*; OUSD(R&E): Washington, DC, USA, 2022.
99. Silva, S.; Tuyishime, A.; Santilli, T.; Pelliccione, P.; Iovino, L. Quality Metrics in Software Architecture. In Proceedings of the IEEE 20th International Conference on Software Architecture Companion (ICSA-C), L'Aquila, Italy, 13–17 March 2023.
100. Torrese, G.; Moriceau, P. An Analytical Approach to Compute the System Architectural Complexity. *IEEE Syst. J.* **2023**, *17*, 3172–3182. [\[CrossRef\]](#)
101. Smagulova, K.; Fouda, M.E.; Kurdahi, F.; Salama, K.N.; Eltawil, A. Resistive Neural Hardware Accelerators. *Proc. IEEE* **2023**, *111*, 500–527. [\[CrossRef\]](#)
102. Vo, Q.H.; Asim, F.; Alimkhanuly, B.; Lee, S.; Kim, L. Hardware Platform-Aware Binarized Neural Network Model Optimization. *Appl. Sci.* **2022**, *12*, 1296. [\[CrossRef\]](#)
103. Li, T.L.; Li, J.D.; Shen, G.B.; Zhao, D.C.; Zhang, Q.; Zeng, Y. FireFly-S: Exploiting Dual-Side Sparsity for Spiking Neural Networks Acceleration With Reconfigurable Spatial Architecture. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2024**, *72*, 4007–4020.
104. Mao, W.D.; Wang, M.Q.; Xie, X.R.; Wu, X.; Wang, Z.F. Hardware Accelerator Design for Sparse DNN Inference and Training: A Tutorial. *IEEE Trans. Circuits Syst. II Express Briefs* **2024**, *71*, 1708–1714. [\[CrossRef\]](#)
105. Soto, J.E.; Hernández, C.; Figueroa, M. JACC-FPGA: A hardware accelerator for Jaccard similarity estimation using FPGAs in the cloud. *Future Gener. Comput. Syst.* **2023**, *138*, 26–42. [\[CrossRef\]](#)
106. Denisov, L.; Galimberti, A.; Cattaneo, D.; Agosta, G.; Zoni, D. Design-time methodology for optimizing mixed-precision CPU architectures on FPGA. *J. Syst. Archit.* **2024**, *155*, 103257. [\[CrossRef\]](#)
107. Tuli, S.; Jha, N.K. EdgeTran: Device-Aware Co-Search of Transformers for Efficient Inference on Mobile Edge Platforms. *IEEE Trans. Mob. Comput.* **2024**, *23*, 7012–7029. [\[CrossRef\]](#)
108. Rios, M.; Ponzina, F.; Levisse, A.; Ansaloni, G.; Atienza, D. Bit-Line Computing for CNN Accelerators Co-Design in Edge AI Inference. *IEEE Trans. Emerg. Top. Comput.* **2023**, *11*, 358–372. [\[CrossRef\]](#)
109. Ling, M.; Feng, Z.H.; Chen, R.Q.; Shao, Y.; Tang, S.D.; Zhu, Y.X. Vina-FPGA-Cluster: Multi-FPGA Based Molecular Docking Tool With High-Accuracy and Multi-Level Parallelism. *IEEE Trans. Biomed. Circuits Syst.* **2024**, *18*, 1321–1337. [\[CrossRef\]](#) [\[PubMed\]](#)
110. Li, C.X.; Fan, X.Y.; Wu, X.T.; Yang, Z.; Wang, M.; Zhang, M.; Zhang, S.B. Memory-Computing Decoupling: A DNN Multitasking Accelerator With Adaptive Data Arrangement. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2022**, *41*, 4112–4123. [\[CrossRef\]](#)

111. Reddi, V.J.; Cheng, C.; Kanter, D.; Mattson, P.; Schmuelling, G.; Wu, C.J. The Vision Behind MLPerf: Understanding AI Inference Performance. *IEEE Micro* **2021**, *41*, 10–18. [\[CrossRef\]](#)
112. Laxmisagar, H.S.; Hanumantharaju, M.C. FPGA implementation of breast cancer detection using SVM linear classifier. *Multimed. Tools Appl.* **2023**, *82*, 41105–41128. [\[CrossRef\]](#)
113. Johnson, J.; Boxer, B.; Prakash, T.; Grace, C.; Sorensen, P.; Tripathi, M. Investigating resource-efficient neutron/gamma classification ML models targeting eFPGAs. *J. Instrum.* **2024**, *19*, P07034. [\[CrossRef\]](#)
114. Schoonhoven, R.A.; van Werkhoven, B.; Batenburg, K.J. Benchmarking Optimization Algorithms for Auto-Tuning GPU Kernels. *IEEE Trans. Evol. Comput.* **2023**, *27*, 550–564. [\[CrossRef\]](#)
115. Crisci, L.; Carpentieri, L.; Cosenza, B.; Accordi, G.; Gadioli, D.; Vitali, E.; Palermo, G.; Beccari, A.R. Enabling performance portability on the LiGen drug discovery pipeline. *Future Gener. Comput. Syst.* **2024**, *158*, 44–59. [\[CrossRef\]](#)
116. Sarker, A.; Canto, A.C.; Kermani, M.M.; Azarderakhsh, R. Error Detection Architectures for Hardware/Software Co-Design Approaches of Number-Theoretic Transform. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2023**, *42*, 2418–2422. [\[CrossRef\]](#)
117. Silva, M.; Gomes, T.; Ekpanyapong, M.; Tavares, A.; Pinto, S. ChamelloT: A tightly- and loosely-coupled hardware-assisted OS framework for low-end IoT devices. *Real-Time Syst.* **2024**, *60*, 150–196.
118. Akgün, G.; Kolarov, B.; Kalberlah, H.; Wulf, C.; Willig, M.; Rettkowski, J.; Göhringer, D. Exploration of Power-Savings on Multi-Core Architectures with Offloaded Real-Time Operating System. *IEEE Access* **2024**, *12*, 11294–11315. [\[CrossRef\]](#)
119. Unal, D.; Falleni, S.; Enhos, K.; Demirors, E.; Basagni, S.; Melodia, T. Design and performance evaluation of SEANet, a software-defined networking platform for the Internet of Underwater Things. *Comput. Netw.* **2024**, *250*, 110579. [\[CrossRef\]](#)
120. Kanso, H.; Noureddine, A.; Exposito, E. Automated power modeling of computing devices: Implementation and use case for Raspberry Pis. *Sustain. Comput. Inform. Syst.* **2023**, *37*, 100837. [\[CrossRef\]](#)
121. Weiss, K.; Rottleuthner, M.; Schmidt, T.C.; Wählich, M. PHiLIP on the HiL: Automated Multi-Platform OS Testing With External Reference Devices. *ACM Trans. Embed. Comput. Syst.* **2021**, *20*, 91. [\[CrossRef\]](#)
122. Kim, J.; Kim, E.; Jeon, S.; Ahn, J.; Jun, H.; Cha, H. PVoT: Reconfigurable Photovoltaic Array for Indoor Light Energy-Powered Batteryless Devices. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2022**, *41*, 4181–4192. [\[CrossRef\]](#)
123. Jiang, Z.; Dai, X.T.; Burns, A.; Audsley, N.; Gu, Z.H.; Gray, I. A High-Resilience Imprecise Computing Architecture for Mixed-Criticality Systems. *IEEE Trans. Comput.* **2023**, *72*, 29–42. [\[CrossRef\]](#)
124. Gaffurini, M.; Flammini, A.; Ferrari, P.; Carvalho, D.F.; Godoy, E.P.; Sisinni, E. End-to-End Emulation of LoRaWAN Architecture and Infrastructure in Complex Smart City Scenarios Exploiting Containers. *Sensors* **2024**, *24*, 2024. [\[CrossRef\]](#) [\[PubMed\]](#)
125. Guevara, L.M.M.; Vidal, P.J.; Olivera, A.C.; Millán, E.N. Analysis and comparison of encryption and verification algorithms applied to a Fog Computing Architecture. *J. Inf. Telecommun.* **2024**, *9*, 173–189. [\[CrossRef\]](#)
126. Horne, R.; Mauw, S.; Mizera, A.; Stemper, A.; Thoemel, J. Anomaly Detection Using Deep Learning Respecting the Resources on Board a CubeSat. *J. Aerosp. Inf. Syst.* **2023**, *20*, 859–872. [\[CrossRef\]](#)
127. Jiang, Z.; Yang, K.C.; Fisher, N.; Audsley, N.; Dong, Z. Towards an energy-efficient quarter-clairvoyant mixed-criticality system. *J. Syst. Archit.* **2022**, *130*, 102683. [\[CrossRef\]](#)
128. Xu, Y.Q.; Li, B.Y.; Zhu, Z.W.; Liu, W.H.; Jia, G.Y.; Han, G.J.; Li, X. MultiRuler: A Multi-Dimensional Resource Modeling Method for Embedded Intelligent Systems of Autonomous Driving. *IEEE Trans. Veh. Technol.* **2024**, *73*, 6212–6224. [\[CrossRef\]](#)
129. Kumar, A.S.; Reddy, B.N.K. An Efficient Real-Time Embedded Application Mapping for NoC Based Multiprocessor System on Chip. *Wirel. Pers. Commun.* **2023**, *128*, 2937–2952.
130. Xu, R.; Sha, E.H.M.; Zhuge, Q.F.; Song, Y.H.; Wang, H.; Shi, L. Optimizing Data Placement for Hybrid SRAM plus Racetrack Memory SPM in Embedded Systems. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2023**, *42*, 847–859. [\[CrossRef\]](#)
131. Nguyen, M.T.; Truong, H.L. On Optimizing Resources for Real-Time End-to-End Machine Learning in Heterogeneous Edges. *Softw. Pract. Exp.* **2025**, *55*, 541–558.
132. Morell, J.A.; Dahi, Z.A.; Chicano, F.; Luque, G.; Alba, E. A multi-objective approach for communication reduction in federated learning under devices heterogeneity constraints. *Future Gener. Comput. Syst.* **2024**, *155*, 367–383. [\[CrossRef\]](#)
133. Kaiser, S.; Tosun, A.S.; Korkmaz, T. Benchmarking Container Technologies on ARM-Based Edge Devices. *IEEE Access* **2023**, *11*, 107331–107347. [\[CrossRef\]](#)
134. Abed, S.; Jaffal, R.; Mohd, B.J.; Al-Shayegi, M. An analysis and evaluation of lightweight hash functions for blockchain-based IoT devices. *Clust. Comput.* **2021**, *24*, 3065–3084. [\[CrossRef\]](#)
135. Beshley, M.; Kryvinska, N.; Beshley, H.; Panchenko, O.; Medvetskyi, M. Traffic engineering and QoS/QoE supporting techniques for emerging service-oriented software-defined network. *J. Commun. Netw.* **2024**, *26*, 99–114. [\[CrossRef\]](#)
136. Kurungadan, B.; Abdrabou, A. Using Software-Defined Networking for Data Traffic Control in Smart Cities with WiFi Coverage. *Symmetry* **2022**, *14*, 2053. [\[CrossRef\]](#)
137. Demirezen, M.U.; Navruz, T.S. Performance Analysis of Lambda Architecture-Based Big-Data Systems on Air/Ground Surveillance Application with ADS-B Data. *Sensors* **2023**, *23*, 7580. [\[CrossRef\]](#) [\[PubMed\]](#)

138. De Simone, L.; Di Mauro, M.; Natella, R.; Postiglione, F. Performance and Availability Challenges in Designing Resilient 5G Architectures. *IEEE Trans. Netw. Serv. Manag.* **2024**, *21*, 5291–5303. [\[CrossRef\]](#)
139. Heideker, A.; Kamienski, C. Network Queuing Assessment: A Method to Detect Bottlenecks in Service Function Chaining. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 4650–4661. [\[CrossRef\]](#)
140. Fu, Y.R.; Zhang, Y.; Zhu, Q.; Dai, H.N.; Li, M.M.; Quek, T.Q.S. A New Vision of Wireless Edge Caching Networks (WECNs): Issues, Technologies, and Open Research Trends. *IEEE Netw.* **2024**, *38*, 247–253. [\[CrossRef\]](#)
141. Ikhlasse, H.; Benjamin, D.; Vincent, C.; Hicham, M. Recent implications towards sustainable and energy efficient AI and big data implementations in cloud-fog systems: A newsworthy inquiry. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 8867–8887. [\[CrossRef\]](#)
142. Fé, I.; Nguyen, T.A.; Di Mauro, M.; Postiglione, F.; Ramos, A.; Soares, A.; Choi, E.; Min, D.G.; Lee, J.W.; Silva, F.A. Energy-aware dynamic response and efficient consolidation strategies for disaster survivability of cloud microservices architecture. *Computing* **2024**, *106*, 2737–2783. [\[CrossRef\]](#)
143. Aslanpour, M.S.; Toosi, A.N.; Cheema, M.A.; Chhetri, M.B.; Salehi, M.A. Load balancing for heterogeneous serverless edge computing: A performance-driven and empirical approach. *Future Gener. Comput. Syst.* **2024**, *154*, 266–280. [\[CrossRef\]](#)
144. Upadhyaya, P.S.; Tripathi, N.; Gaeddert, J.; Reed, J.H. Open AI Cellular (OAIC): An Open Source 5G O-RAN Testbed for Design and Testing of AI-Based RAN Management Algorithms. *IEEE Netw.* **2023**, *37*, 7–15. [\[CrossRef\]](#)
145. Xu, T.N.; Sun, H.F.; Zhang, D.; Zhou, X.M.; Sui, X.F.; Wang, S.; Huang, Q.; Bao, Y.G. NfvInsight: A Framework for Automatically Deploying and Benchmarking VNF Chains. *J. Comput. Sci. Technol.* **2022**, *37*, 680–698. [\[CrossRef\]](#)
146. Zhang, Q.; Liu, Y.; Liu, T. iBalancer: Load-Aware in-Server Flow Scheduling for Sub-Millisecond Tail Latency. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 1761–1774. [\[CrossRef\]](#)
147. Sheraz, S.M.; Arfeen, A.; Haider, U. Energy-Efficient Data Center Network Infrastructure With Network Switch Refresh Model. *IEEE Access* **2023**, *11*, 45066–45082. [\[CrossRef\]](#)
148. Zhu, Y.Y.; Wei, S.J.; Tiwari, M. Revisiting Browser Performance Benchmarking From an Architectural Perspective. *IEEE Comput. Archit. Lett.* **2022**, *21*, 113–116. [\[CrossRef\]](#)
149. Kohyarnejadfar, I.; Aloise, D.; Dagenais, M.R.; Shakeri, M. A Framework for Detecting System Performance Anomalies Using Tracing Data Analysis. *Entropy* **2021**, *23*, 1011. [\[CrossRef\]](#) [\[PubMed\]](#)
150. Zaghloul, M.; Salem, M.; Ali-Eldin, A. A new framework based on features modeling and ensemble learning to predict query performance. *PLoS ONE* **2021**, *16*, e0258439. [\[CrossRef\]](#) [\[PubMed\]](#)
151. Arroqui, M.; Ferreira, F.; Roselli, M.; Alvarez, J.R.; Machado, C.; Mateos, C.; Zunino, A. A Simulation Scheduling Module to Improve User Experience in the Simugan Beef-Cattle Farm Simulator. *IEEE Lat. Am. Trans.* **2022**, *20*, 162–170. [\[CrossRef\]](#)
152. ISO/IEC 25010:2011; Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—System and Software Quality Models. International Organization for Standardization: Geneva, Switzerland, 2011.
153. Liu, W.H.; Zhu, Z.W.; Li, B.Y.; Xiong, Y.; Lian, Z.R.; Geng, J.W.; Zhou, X.H. Arch2End: Two-Stage Unified System-Level Modeling for Heterogeneous Intelligent Devices. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2024**, *43*, 4154–4165. [\[CrossRef\]](#)
154. Mazumder, A.N.; Mohsenin, T. Reg-TuneV2: A Hardware-Aware and Multiobjective Regression-Based Fine-Tuning Approach for Deep Neural Networks on Embedded Platforms. *IEEE Micro* **2023**, *43*, 74–83. [\[CrossRef\]](#)
155. Pazzaglia, P.; Maggio, M. Characterizing the Effect of Deadline Misses on Time-Triggered Task Chains. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2022**, *41*, 3957–3968. [\[CrossRef\]](#)
156. Tirelli, C.; Sapriza, J.; Alvarez, R.P.; Ferretti, L.; Denking, B.; Ansaloni, G.; Calero, J.M.; Atienza, D.; Pozzi, L. SAT-based Exact Modulo Scheduling Mapping for Resource-Constrained CGRAs. *ACM J. Emerg. Technol. Comput. Syst.* **2024**, *20*, 8. [\[CrossRef\]](#)
157. Liu, H.Z.; Luo, J.; Li, Y.; Wu, Z.H. Iterative Compilation Optimization Based on Metric Learning and Collaborative Filtering. *ACM Trans. Archit. Code Optim.* **2022**, *19*, 2.
158. Wang, C.; Gong, L.; Li, X.; Yu, Q.; Wang, A.L.; Hung, P.T.; Zhou, X.H. SOLAR: Services-Oriented Deep Learning Architectures-Deep Learning as a Service. *IEEE Trans. Serv. Comput.* **2021**, *14*, 262–273.
159. Memis, S.; Yeniceri, R. Model-Based FPGA Implementation of a 6-DoF Dynamical Model Accelerator. *IEEE Access* **2024**, *12*, 45279–45298. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.