1

Edge AI System-of-Systems Reference Architecture Engineering Foundations and Multi-Dimensional Views

Ovidiu Vermesan[1], Marcello Coppola[2], Silke Braun[3], and Patrick Pype[4]

[1]SINTEF AS, Norway
[2]STMicroelectronics, France
[3]Infineon Technologies Austria AG, Austria
[4]NXP Semiconductors, Belgium

Abstract

Edge AI systems are emerging from the convergence of IoT, edge computing, AI, agentic AI, and embodied, physical generative edge AI delivering adaptive, autonomous behaviour under physical, cyber, and operational constraints while remaining trustworthy. This article frames edge AI as a complex system-of-systems in which hardware, software, models, and data continuously co-evolve across heterogeneous "multi-X" environments: multiple systems, modalities, and agents distributed from the edge to the cloud. The article argues that as edge AI technologies are maturing, there is a need for a standardised, application-agnostic reference architecture to provide a shared lexicon and taxonomy, reduce integration errors, and expose opportunities for reusable assets and productive interoperability and standardisation. The paper grounds this need in systems engineering and introduces a quad-optimisation paradigm for balancing competing objectives during design and operation. The article presents a design framework and a multi-dimensional architecture organised into three complementary views: quality properties for trustworthiness and dependability, a layered technology stack within each tier, and a processing continuum that partitions intelligence across edge-to-cloud tiers. Finally, the article discusses value creation, interoperability, and how a

common baseline supports the development of complex edge AI systems-of-systems and their verification, validation, testing and benchmarking.

Keywords: edge AI, edge AI systems reference architecture, generative edge AI, agentic edge AI, embodied edge AI, physical edge AI.

1.1 Edge AI as a Complex System-of-Systems

Edge AI systems represent a class of engineered systems arising from the deep convergence of the Internet of Things (IoT), edge computing paradigms, Artificial Intelligence (AI), generative AI (GenAI), AI agents, and agentic AI architectures as represented in Figure 1.1. Edge AI systems operate continuously under stringent physical, cyber, and operational constraints while remaining highly adaptive, autonomous, and thoroughly trustworthy [14].

From a rigorous systems engineering perspective, this technological convergence produces a highly complex system-of-systems (SoS). Within this SoS, hardware modules, software components, deployed AI models, and real-time data are tightly coupled and continuously evolving together.

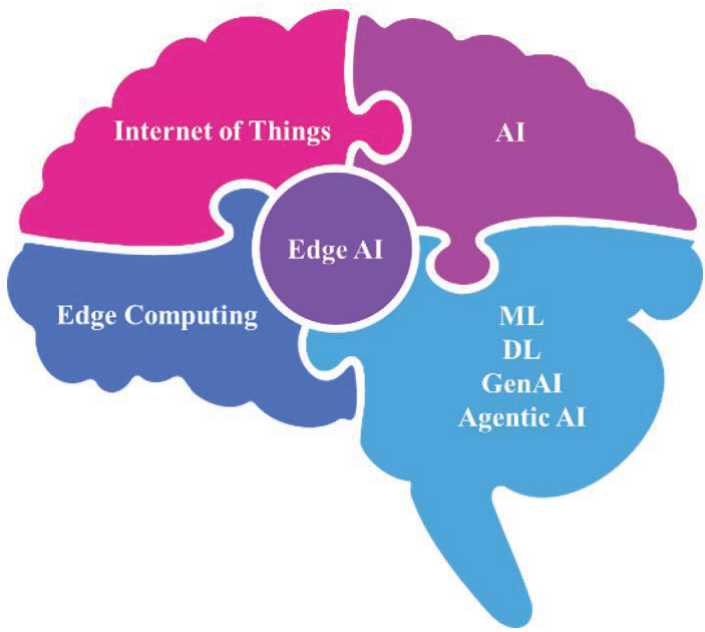

Figure 1.1 Edge AI as a complex system-of-systems.

This intrinsic, compounding complexity creates an interlinked scientific and engineering rationale for developing a standardised, application-agnostic reference architecture tailored for edge AI environments.

Traditional systems engineering principles emphasise abstraction, clear separation of concerns, requirements traceability, and comprehensive lifecycle thinking. However, edge AI systems actively challenge these traditional principles because critical functionality and system-quality properties emerge dynamically from complex interactions across distributed components rather than originating from isolated, static subsystems.

For instance, architectural decisions at the silicon hardware level directly and in real-time influence the feasibility, energy efficiency, inference latency, and overall system reliability of edge AI models. Concurrently, data governance policies and lifecycle management protocols for machine learning (ML) or deep learning (DL) models profoundly affect the system's trustworthiness and compliance with stringent regulatory frameworks. A robust reference architecture provides a stable, formalised conceptual structure that makes these intricate cross-layer interactions explicitly visible and analysable. This structural clarity enables systematic engineering trade-off analysis and robust system design, actively preventing brittle, ad hoc integrations.

In this context, this conceptual article proposes a foundational framework intended to guide and shape future architectural work. Because its primary purpose is to establish a theoretical blueprint for strategic design, it does not rely on empirical data or require experimental validation. Instead, the principles and structural concepts outlined herein are broadly applicable to upcoming development initiatives, providing teams with a cohesive, forward-looking vision to direct their long-term technical decisions.

The article's structure is briefly introduced below. The article reviews reference-architecture concepts from systems engineering as a foundation for standardisation and clarifies the purpose of an application-agnostic edge AI systems reference architecture and the value it enables. The article introduces the quad-optimisation paradigm to manage trade-offs across performance, resources, trust, and lifecycle evolution. It outlines a practical framework for designing the reference architecture across projects and disciplines. The following subsections present the Quality Properties View to engineer trustworthiness and dependability above the single-system level and detail the Technology Stack and Processing Continuum views to align tiers, layers, and intelligence partitioning. The article concludes by discussing value, interoperability, domain maturity, and the architecture's role as a baseline for edge AI systems designers and stakeholders

1.2 The Foundations of Reference Architectures in Systems Engineering

A reference architecture serves as an authoritative blueprint for a specific subject or domain. This foundational document actively guides and constrains the creation of multiple subsequent architectures and concrete solutions [19]. By providing a rigorous framework of technical standards and engineering rules, a reference architecture ensures that implementations across a complex domain remain highly repeatable, deeply interoperable, and fundamentally consistent. Reference architectures explicitly define the necessary conditions to achieve overarching domain goals and strategic objectives. Consequently, downstream solutions describe precisely how to achieve these goals as real-world implementations. These solution architectures include the concrete, specific details regarding the processes and computational resources required to deliver critical missions, system capabilities, and technological services.

Reference architectures do not exist in isolation; they are complemented by architecture frameworks. These frameworks provide structured guidance and rules for classifying and organising complex system information. They consist of an organised set of layered, hierarchical artefacts encompassing architectural descriptions, stakeholder perspectives, visualisations, and fundamental building blocks. Furthermore, these frameworks define how architectural data elements fit together and relate within the broader system. Reference models are subsequently utilised to construct these architectures, representing formalised taxonomies that provide standardised categorisation of system entities.

There is no "one-size-fits-all" approach to systems architecture as the systems must address the users' physical constrains and environmental factors as illustrated in Figure 1.2. Because every technological domain possesses unique characteristics and operational constraints, reference architectures naturally vary in scope, levels of abstraction, and overall coverage. Each proposed architecture must be stringently "fit-for-purpose," designed specifically to maximise functional value and accelerate technological development within its intended domain.

When designing a robust reference architecture, several core components must be meticulously addressed. The architecture must clearly identify its fundamental purpose, specifically outlining the goals, objectives, and engineering problems it intends to solve. It must specify its principles, establishing the high-level engineering foundations that drive technical positions

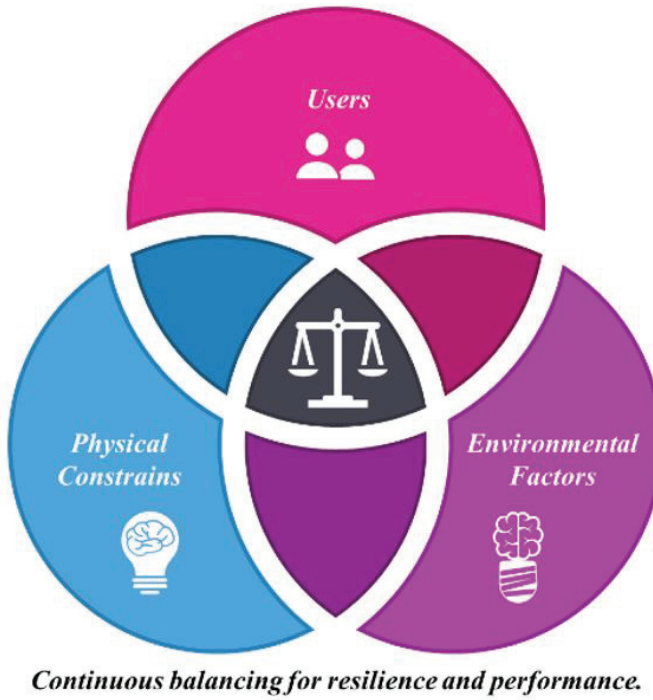

Figure 1.2 Edge AI systems boundaries.

and structural patterns. Furthermore, it must define technical positions by integrating standards, policies, and communication protocols to constrain solutions and ensure regulatory compliance. The architecture must also provide general architectural patterns, unconstrained by specific implementation details, to guide development. Finally, establishing a standardised vocabulary through a comprehensive glossary of terms ensures consistent communication among diverse engineering teams [2].

Consequently, a complete and mature reference architecture encompasses a clearly articulated goal and problem space, detailing the recurring problem, operational context, and intended use cases. It defines strict scope and boundaries, establishing the required level of granular detail while explicitly stating what remains outside the architecture's purview. It establishes principles and guidelines for deploying specific technologies, forming a rational basis for future technical decision-making. The architecture identifies reusable components and their intricate relationships across logical, process, physical, and scenario-based views. Finally, it inherently incorporates industry best

practices, integrating accepted external standards and common architectural patterns [2, 21].

Standards such as ISO 15704:2019 [11], ISO/IEC/IEEE DIS 42024 [7] and ISO/IEC/IEEE DIS 42042 [9] address structuring architectural knowledge in a clear, consistent, and reusable way and emphasise that a reference architecture should not be just a diagram, but a well-defined set of concepts, relationships, and rules that guide system design across multiple implementations. A key common element is the use of viewpoints and views. Each standard recognises that stakeholders have distinct concerns, so architectures must be described from multiple perspectives. This ensures that business, functional, information, and technical aspects are all addressed in a coordinated manner. The standards align in defining the importance of concepts, constructs, and relationships. Whether framed as enterprise constructs in ISO 15704:2019 [11] or architectural elements in the ISO/IEC/IEEE standards, the standards stress the need for a formal vocabulary and meta-model to avoid ambiguity and enable interoperability. The standards acknowledge that architectures evolve over time and must support phases such as design, implementation, operation, and evolution.

The conceptualisation of a system's architecture, as defined in ISO/IEC/IEEE 42010:2022 [5], "Systems and software engineering – Architecture," helps understand the system's essence and key properties related to its behaviour and composition. The standard specifies requirements for the structure and expression of an architecture description (AD) for various entities, including software, systems, enterprises, systems of systems, families of systems, products (goods or services), product lines, service lines, technologies, and business domains. The document distinguishes the architecture of an entity of interest from an AD that expresses that architecture and specifies requirements for the use of architectural concepts and their relationships as captured in an AD. It does not specify requirements for any entity of interest or its environment.

The concept described in the standard can support the first steps in defining an edge AI systems reference architecture, as the standard specifies in general requirements for an architecture description framework (ADF), an architecture description language (ADL), architecture viewpoints, and model kinds to usefully support the development and use of an AD. It describes the system's structure, including its entities, and the interactions between each entity and the environment.

ISO/IEC/IEEE 42010:2022 [5] considers that an architecture description expresses the architecture of a system of interest, in this case, an edge AI

system. While an architecture can be abstract, consisting of concepts and properties, an AD is a work product formalising an architecture, including one or more architecture views. An architecture view addresses one or more concerns of the edge AI system's stakeholders.

An architecture view expresses the architecture of the system of interest from a given architecture viewpoint. As a result, the architecture framework (AF) contains the conventions, principles, and practices for describing architectures established within a specific domain of application and/or community of stakeholders. The AF can be described along several dimensions, establishing conventions for the construction, interpretation, and use of a system's architecture from the perspective of specific system concerns.

A graphical representation of the entities described, and their interrelationships is presented in Figure 1.3.

The lifecycle perspective ensures that reference architectures remain relevant and adaptable. Consistency and traceability are also central. Each of the ISO 15704:2019 [11], ISO/IEC/IEEE DIS 42024 [7] and ISO/IEC/IEEE DIS 42042 [9] standards requires that architectural decisions, elements, and views be logically connected, allowing stakeholders to trace requirements through implementation and assess impacts of change. All these standards promote reusability and governance. Reference architectures are intended to provide

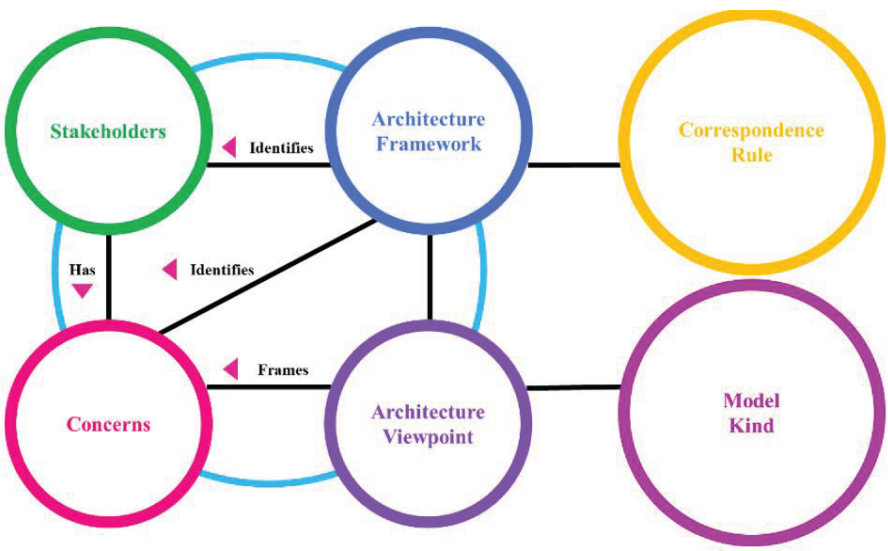

Figure 1.3 Graphical representation of entities and their interrelationships (Adapted from ISO/IEC/IEEE 42010:2022 [5]).

reusable patterns, constraints, and best practices, while also supporting governance mechanisms that ensure compliance and alignment across systems and projects.

1.3 Purpose of an Edge AI Systems Reference Architecture

Systems engineering principles emphasise abstraction, separation of concerns, traceability, and lifecycle thinking. Edge AI systems challenge these principles because functionality and quality properties emerge from interactions across distributed components rather than from isolated subsystems. Decisions made at the hardware level directly influence the feasibility, energy efficiency, latency, and reliability of edge AI models, while data governance and model lifecycle management affect trustworthiness and regulatory compliance. An edge AI systems reference architecture provides a stable conceptual structure that makes these interactions explicit and analysable, enabling systematic trade-off analysis and robust design rather than ad hoc integration.

An edge AI systems reference architecture provides a shared, application-agnostic architectural baseline for engineering, integrating, and evolving edge AI systems in the presence of system heterogeneity. Edge deployments differ widely in compute and memory budgets, sensors and modalities, network connectivity, safety and security exposure, energy constraints, and lifecycle practices. Without a common architectural frame, each solution tends to develop its own terminology, assumptions, and interfaces, making cross-application communication, integration, and reuse inefficient and error prone.

This need becomes more acute as edge AI systems shift from relatively simple products to true systems-of-systems implemented in autonomous systems. Modern deployments increasingly comprise multiple edge nodes, multiple edge AI-enabled functions, multiple stakeholders, and multiple operational domains, with intelligence distributed across edge-to-cloud tiers. The resulting coupling across application boundaries and technical layers makes it difficult to maintain coherence, manage change, and reason about emergent behaviour unless there is an explicit architectural structure that can be shared and governed.

Edge AI systems-of-systems also combine multiple edge AI methods and interaction patterns within a single operational envelope. Machine learning (ML), deep learning (DL), generative AI, and agentic edge AI frequently

coexist, interact, and co-adapt, often with different data requirements, safety implications, explainability needs, and runtime resource profiles. A reference architecture makes these method families key concerns, enabling consistent placement of responsibilities (for example, where generation is allowed, where agents can act, and where constraints are enforced) and supporting disciplined partitioning of intelligence across tiers.

A defining purpose is to enable co-design and optimisation across hardware, software, AI, and data as a single engineered whole. At the edge, architectural choices such as model form, quantisation strategy, scheduling, accelerators, memory hierarchy, data pipelines, and privacy/security controls are tightly interdependent. A reference architecture establishes the canonical set of architectural elements, interfaces, and viewpoints needed to jointly optimise these dimensions, rather than optimising each element in isolation and discovering conflicts late during integration or operation.

Another core purpose is to provide a common reference for comparing edge AI systems in verification, validation, testing, and benchmarking. Edge AI quality is multi-dimensional, spanning functional correctness, timing predictability, robustness, safety, security, privacy, resilience, and maintainability under continuous updates. A reference architecture supports comparable claims by standardising how systems are described, what constitutes evidence, which qualities are assessed at which level (component, subsystem, system, and SoS), and how assumptions and constraints are recorded. This directly reduces ambiguity in test scope, improves traceability from requirements to evidence, and enables fair benchmarking across heterogeneous platforms and workloads.

These purposes align with the intent of TOGAF's principles [20, 21], which frame architecture as a mechanism to translate drivers into a governed, coherent set of building blocks and implementation guidance. In TOGAF's terms, a reference architecture supports stakeholder alignment by providing a stable target architecture pattern that projects can tailor while remaining interoperable. It improves decision-making by making trade-offs explicit early, informs migration planning by clarifying what can be standardised versus what must vary, and enables reuse through a consistent catalogue of architectural building blocks, patterns, and interfaces across a portfolio of edge AI solutions [20].

They also align with ISO/IEC/IEEE 42010:2022 [5], which emphasises architecture description to address stakeholder concerns through viewpoints, views, and documented rationale. For edge AI, the reference architecture's purpose is to make concerns such as trustworthiness, dependability (including

safety and security), data governance, and lifecycle evolution explicit and analysable, rather than implicit and scattered across implementations. By standardising viewpoints and terminology, it strengthens communication across disciplines, supports the analysis of alternatives and impacts, and provides a basis for consistency checking across systems and over time as systems evolve.

As a result, an edge AI systems reference architecture exists to manage heterogeneity and SoS complexity, to structure the co-design of hardware–software–AI–data under edge constraints, and to provide a standardised basis for communication, governance, comparison, and evidence-driven assurance. It is the shared starting point that makes scalable engineering, interoperability, and credible verification, validation, testing and benchmarking achievable as the edge AI domain matures.

1.4 The Paradigm of Quad-Optimisation

The absolute requirement for continuous, simultaneous optimisation across hardware, software, the technology stack, and data pipelines, referred to as quad-optimisation, further drives the critical need for a standardised reference edge AI systems architecture, as illustrated in Figure 1.4 [14].

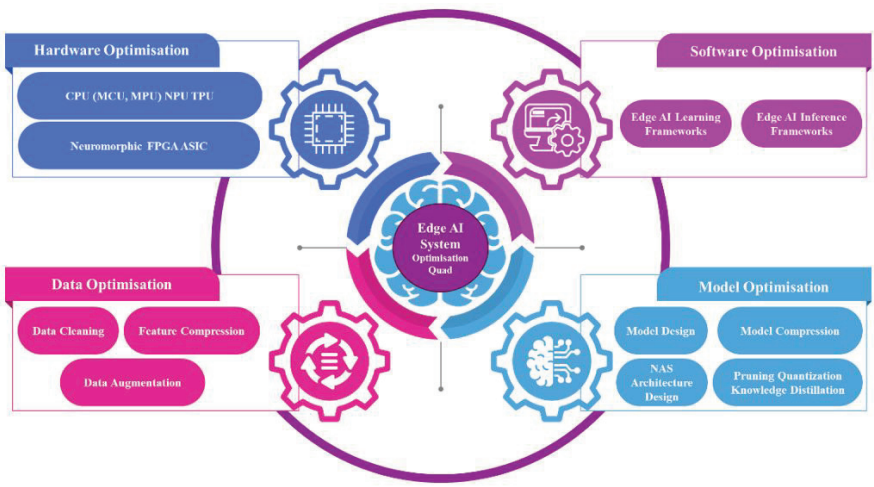

Figure 1.4 Edge AI quad optimisation.

In constrained edge environments, systemic optimisation objectives are often in conflict with one another. Engineers must constantly navigate trade-offs such as edge AI model accuracy versus physical energy consumption, inference latency versus model explainability, or data privacy versus high availability. Crucially, these complex trade-offs cannot be resolved locally within a single architectural layer or isolated component. A comprehensive reference architecture deliberately embeds quad optimisation as a first-class architectural concern. This embedding allows system designers to logically reason about structural co-evolution and dynamic adaptation across the full system lifecycle, encompassing initial deployment, continuous operation, remote monitoring, and over-the-air updates for edge AI-driven systems.

This principle of quad-optimisation can be clearly exemplified by the engineering requirements of a battery-powered wildlife edge camera. This system imposes a significant operational trade-off between the edge AI model's detection accuracy and the underlying hardware's energy efficiency. System designers might intentionally select a heavily quantised neural network over a full-precision model, strategically accepting a minor percentage drop in animal detection accuracy. This precise trade-off enables the inference software to run on a significantly smaller, less expensive microcontroller with vastly reduced RAM requirements. This architectural decision extends the remote device's battery life from mere weeks to several months, successfully satisfying the critical sustainability quality property.

The identical engineering principle applies to a collaborative robotic arm designed to work in close physical proximity to human operators. Here, system architects must delicately balance high-resolution data fidelity with strict edge AI system latency limits. The edge AI system might be deliberately configured to process low-resolution, monochrome video data streams rather than high-definition colour streams. By intentionally reducing data richness, the edge AI algorithm becomes computationally simpler, and the software processing time decreases exponentially. This specific architectural optimisation allows the edge hardware to execute emergency safety stops in microseconds, thereby prioritising the critical quality property of human safety over unnecessary image detail.

ISO/IEC/IEEE 42010:2022 provides a foundational, conceptual framework for scientifically describing and modelling complex edge AI systems. The standard formalises the critical technical distinction between a system's physical architecture and its theoretical architectural description. Furthermore, it introduces the vital concepts of system stakeholders, engineering concerns, viewpoints, and architectural views.

In the specific context of edge AI, this standardised framework is essential. Different system stakeholders, including operational system managers, silicon hardware engineers, software developers, AI algorithm designers, embedded systems experts, data scientists, cybersecurity specialists, and domain experts, have highly distinct and often competing engineering concerns. A reference architecture structured strictly according to ISO/IEC/IEEE 42010:2022 [5] explicitly addresses these multifaceted concerns through well-defined, standardised architectural views. This formalised approach guarantees structural consistency, requirements traceability, and clear engineering communication across diverse technical disciplines and application domains.

In practice, quad-optimisation is applied by deeply integrating it into the entire development lifecycle of an edge AI system. Rather than acting as a single, isolated step, it functions as a continuous, iterative process that systematically increases performance outcomes across all phases of development. This optimisation loop begins at the very foundation with requirements engineering and system design, subsequently driving the specific architectural choices for hardware (HW), software (SW), AI models, and data pipelines. This integrated approach extends well beyond the initial design phases; the quad-optimisation framework becomes the standard metric for guiding the rigorous verification, validation, testing, and benchmarking of complex, distributed edge AI systems of systems.

1.5 Methodology and Framework for Edge AI Systems Reference Architecture Design

A rigorous methodology for establishing architecture principles follows a structured, cyclical process. This systematic approach begins by identifying the underlying key business requirements and architectural drivers that necessitate the formulation of specific engineering principles. These fundamental drivers provide the empirical rationale for development and ensure tight alignment with broader organisational strategy and governance objectives.

Three distinct stream activities, assess, aim, and act [3], can be effectively combined with generic system development processes. This integration is generally regarded as a practical realisation of the generalised structural lifecycle required to define complex reference architectures [1], as illustrated in Figure 1.5.

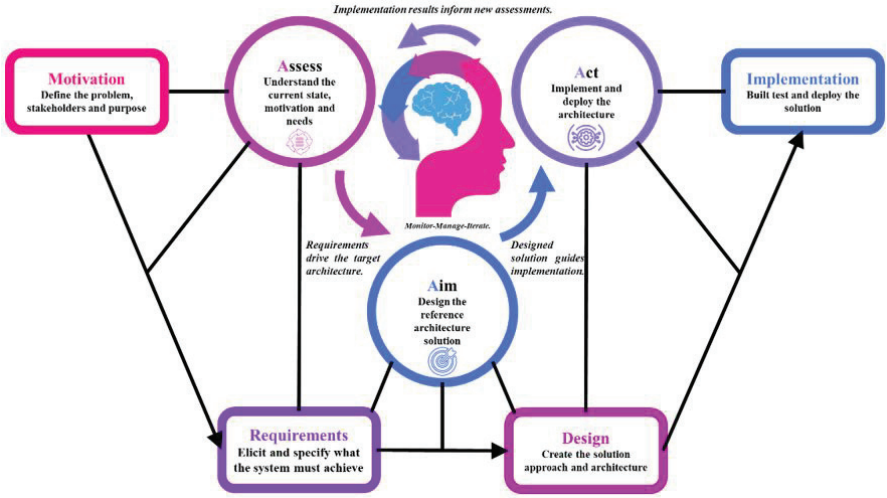

Figure 1.5 Stream of activities used for defining a reference architecture.

The architecture principles, acting as the foundation for best practices based on this stream-of-activities approach, are further refined in subsequent stages. These sub-processes determine, specify, classify, validate, and practically apply the architecture principles. The following sub-process utilises these established principles to definitively determine whether ongoing engineering activities comply with the intended architecture [2]. The final sub-process manages architecture changes and lifecycle iterations, which may inevitably necessitate restarting the initial assessment sub-process.

The assess, aim, and act streams of activities are iterative and cyclically dependent, actively supporting the continuous improvement of the reference architecture. During assessment activities, engineers identify the precise technical motivation for the reference architecture while meticulously gathering the rigorous requirements for a possible solution. These requirements serve as direct input to the aim process, wherein a reference architecture solution is theoretically designed to meet them. The physical implementation, including the practical deployment of the architecture, is subsequently addressed by the act process. The technical requirements specify the measurable properties the reference architecture should possess, and the underlying motivations explain why diverse stakeholders require them. The final design directly reflects how an implemented reference architecture successfully meets these stringent requirements.

A comparison with the TOGAF Architecture Development Method (ADM) [20], the ADM provides a highly specific, standardised way to implement these assess, aim, and act processes. The ADM's architecture vision phase focuses intensely on understanding the essential engineering problem and generating a solution vision, effectively serving as a primary assess/aim iteration. The technology architecture phase provides further granular assess/aim iterations. Depending on the specific situational context, the engineering focus shifts between understanding the problem constraints (assess) and developing the structural solution (aim). Identifying opportunities, generating solutions, and conducting migration planning yield further vital iterations of the aim process, continuously elaborating on the actual intended reference architecture. Finally, the implementation, governance, and architecture change management phases, including the physical realisation of the envisaged architecture, directly correspond to the foundational act process.

Designing a resilient edge AI system is a demanding engineering task, as it requires continually balancing user needs, physical constraints, and environmental factors that often conflict with one another. The design process for edge AI systems invariably begins with selecting the optimal architectural style, also referred to as an architectural pattern, best suited to the system's explicit operational requirements.

An architectural style comprises a set of architectural patterns that share distinct technical characteristics, providing general engineering guidance for reliably solving a specific class of problems within a particular operational context. Each selected architecture style focuses on harmonising and optimising the interplay between hardware, software, selected AI methods, algorithms, software frameworks, operational data, and training datasets.

The comprehensive approach utilised to develop the edge AI systems reference architecture included identifying highly common engineering activities across different industrial value chains. It further involved precisely determining how these disparate functional components are seamlessly combined to create a viable edge AI solution. This was done with the core understanding that the resulting reference architecture must illustrate the logical generalisation of multiple distinct, successful solutions and implementations across different industrial sectors. Foundational reference models that support understanding the structural hierarchy of edge AI systems provide the necessary basics of these systems. These models vary in technical detail, successfully identifying the core architectural elements required for operation.

The rigorous methodology employed for constructing such a reference architecture integrates established architectural description principles with proven, concrete concepts from both edge computing [22] and 3D IoT [15] architectural paradigms as illustrated in Figure 1.6.

The relevance of this methodology is applied across diverse industrial domains, including advanced manufacturing, digital industry, energy, precision agroindustry, beverage, intelligent mobility, and broader digital society applications. In these industrial contexts, rapid, real-time data preprocessing, robust local autonomy, and fail-safe decision-making are essential. Within these domains, a reference edge AI systems architecture functions as a primary scientific and engineering instrument. It effectively structures inherent system complexity, actively supports experimental reproducibility, and enables the systematic, safe evolution of highly scalable, secure, and computationally efficient edge AI systems grounded entirely in recognised international standards.

The edge AI systems reference architecture was conceptualised and designed in accordance with the foundational engineering principles and technical definitions established in major international standards.

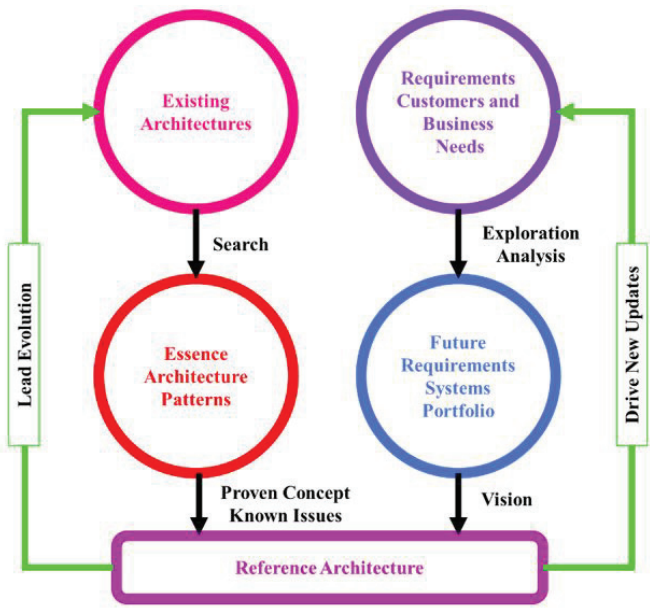

Figure 1.6 Reference architecture methodology development.

These include ISO/IEC/IEEE 42010:2022 [5], ISO/IEC/IEEE 42020:2019 [6], ISO/IEC/IEEE DIS 42024 [7], ISO/IEC/IEEE 42030:2019 [8], ISO/IEC/IEEE DIS 42042 [9], ISO/IEC/IEEE 15288:2023 [10], ISO/IEC 25010:2023 [12], and the comprehensive TOGAF® Standard 10th edition. This alignment and adherence to standardised, globally recognised frameworks ensure that the resulting architecture provides a robust, scientifically sound foundation for edge AI system development that consistently meets modern engineering expectations.

From a pure technology perspective, the architecture dynamically synthesises proven engineering concepts by systematically mining and logically generalising prior industry experience. It draws from established frameworks such as the Industrial Internet Reference Architecture (IIRA) [4], the structured 3D IoT Layered Architecture [15], and the Reference Architecture Model for Edge Computing (RAMEC) [22].

1.5.1 Industrial Internet Reference Architecture (IIRA)

IIRA [4] shown in Figure 1.7 is a standards-based, open architectural framework designed to guide the design and integration of Industrial Internet of Things (IIoT) systems. It is intentionally abstract and generic, allowing it to be applied across diverse industrial domains such as manufacturing, energy, transportation, and healthcare. Rather than prescribing specific technologies or products, IIRA provides a common language and structure for describing complex systems.

At its foundation, IIRA adopts the ISO/IEC/IEEE 42010:2022 standard for architecture description. This standard defines how architectures should be expressed, including the use of viewpoints, stakeholders, and concerns. By aligning with this model, IIRA establishes a consistent ontology for describing IIoT systems, enabling architects, engineers, and stakeholders to communicate using shared concepts and conventions.

The architecture is organised around four primary viewpoints, each addressing a different set of concerns.

The business viewpoint focuses on stakeholders, their objectives, and the value the system is expected to deliver. It captures business goals, regulatory considerations, and economic drivers, ensuring that technical decisions align with organisational priorities.

The usage viewpoint describes how the system is expected to be used in practice. It models interactions as sequences of activities involving human users, automated systems, or both. This viewpoint helps clarify operational

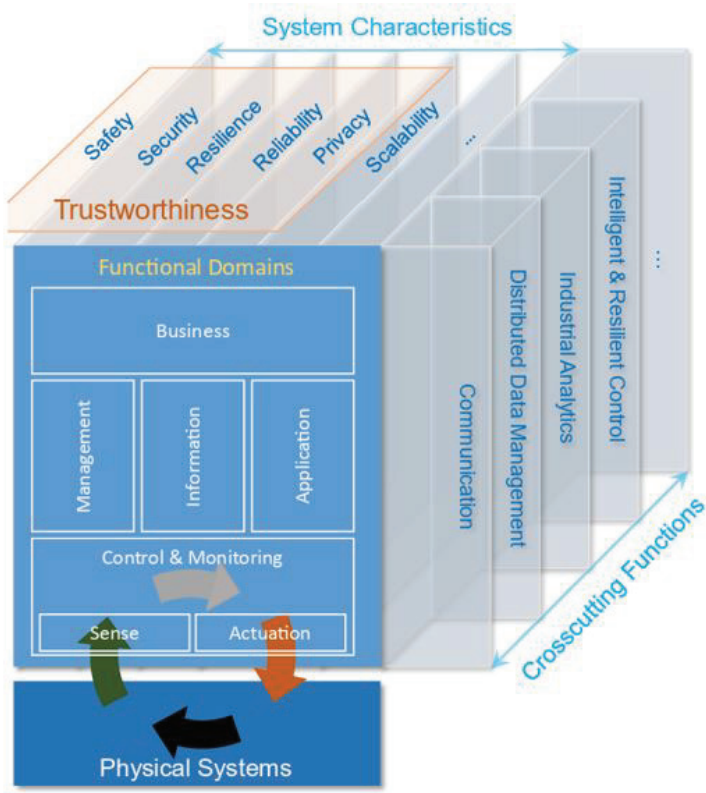

Figure 1.7 IIRA - Functional domains, crosscutting functions and system characteristics.

scenarios, workflows, and system behaviours from an end-user or operational perspective.

The functional viewpoint provides a structural decomposition of the system into functional components and defines how these components interact. It specifies interfaces, data flows, and responsibilities without committing to specific technologies. This abstraction allows designers to reason about system capabilities independently of implementation details.

The implementation viewpoint maps the functional components to concrete technologies, platforms, and deployment configurations. It addresses concerns such as hardware, software frameworks, communication protocols, and integration mechanisms required to realise the system in practice.

In addition to these viewpoints, IIRA introduces two important dimensions that extend across the functional model. System characteristics describe

emergent properties arising from component interactions, such as safety, security, reliability, and scalability. These are not tied to a single component but must be considered holistically throughout the architecture.

Crosscutting functions, closely related to system characteristics, represent supporting capabilities required across multiple functional components. Connectivity is a key example, enabling communication between distributed elements. While the terminology sometimes overlaps with system characteristics, both concepts emphasise concerns that span the entire architecture rather than belonging to a single layer or module.

IIRA provides a structured yet flexible framework that helps organisations design interoperable, scalable, and robust IIoT systems while maintaining alignment between business goals and technical implementation.

The IIRA contributes to the development of the edge AI systems reference architecture by providing the concept of system characteristic elements as part of its foundational design. The architecture introduces a "trustworthiness view" that serves as a reference for evaluating and ensuring overall system integrity. By embedding key characteristics into this view, namely safety, security, resilience, reliability, privacy, and scalability, the systems designed and implemented are functional, robust and dependable across industrial environments.

1.5.2 3D IoT Layered Architecture

The 3D IoT Reference Architecture [15] illustrated in Figure 1.8 extends the traditional layered IoT model by adding two extra dimensions to better capture how real-world IoT systems behave and are engineered. Instead of viewing IoT purely as a stack of functional layers, this approach frames the system as a three-dimensional structure where layers, cross-cutting functions, and system properties interact.

The first dimension consists of the traditional horizontal layers, which represent the core functional decomposition of IoT systems. These layers typically include device, connectivity, data processing, and application levels, each responsible for specific operations. This layered structure reflects how data flows from physical devices up to user-facing services and how control flows in the opposite direction.

The second dimension introduces cross-cutting functions that span multiple layers rather than belonging to a single layer. These functions address concerns such as security, privacy, identity management, and interoperability. Because these concerns must be consistently enforced across the entire

Figure 1.8 3D IoT layered architecture.

system, they cannot be isolated within a single layer. For example, security must be embedded at the device level, maintained during communication, and enforced in data processing and applications.

The third dimension focuses on system properties, which describe the overall qualities of the IoT system. These properties include trustworthiness, reliability, scalability, and performance. They emerge from the proper implementation and interaction of both layered functions and cross-cutting concerns. Trustworthiness is strongly influenced by how effectively security and privacy mechanisms are integrated across all layers.

Together, these three dimensions provide a more comprehensive architectural view. The model allows designers to analyse IoT systems not only in terms of functional decomposition but also in terms of system-wide qualities and transversal concerns. This multidimensional perspective supports better design decisions, ensuring that critical aspects such as security and system reliability are consistently addressed throughout the architecture rather than treated as afterthoughts.

The 3D IoT layered architecture brings important elements to the edge AI systems reference architecture by introducing the concept of cross-cutting functions that span multiple tiers rather than being confined to a single

layer. This multi-dimensional approach recognises that key concerns, such as security, privacy, identity management, and interoperability, cannot be isolated within an individual layer without creating vulnerabilities; instead, they must be consistently enforced across the entire system. By providing structural input through its third dimension, the architecture adds a system properties dimension, ensuring that the overall qualities and integrity of the IoT ecosystem are maintained from edge devices to the cloud embedded into the edge AI systems reference architecture.

1.5.3 Reference Architecture Model for Edge Computing (RAMEC)

RAMEC [22] presented in Figure 1.9 provides a structured way to describe and analyse edge systems by organising them across three complementary viewpoints: cross-layer concerns, hierarchy levels, and technology layers.

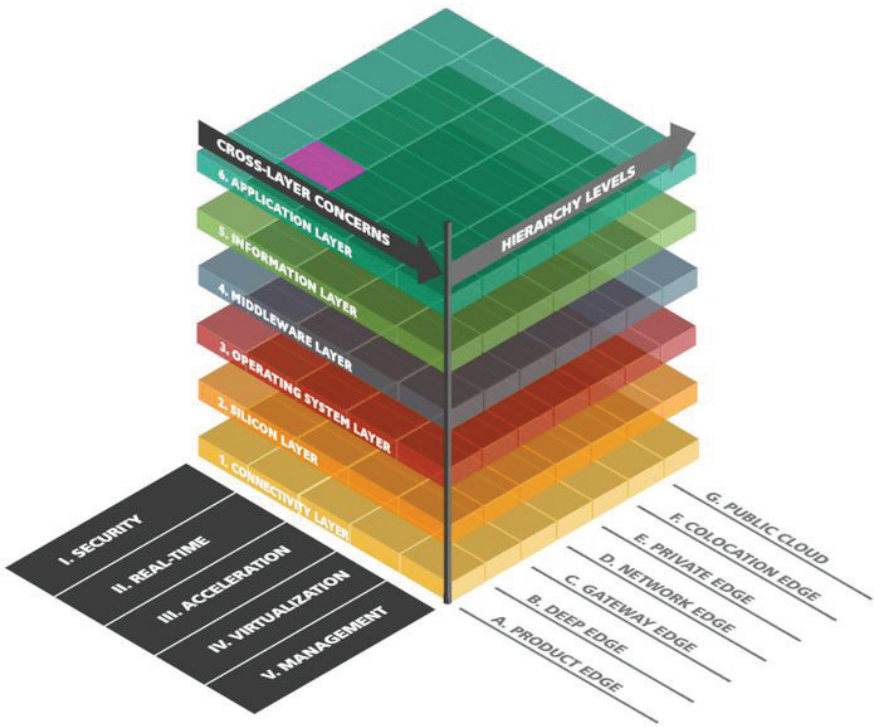

Figure 1.9 Reference architecture model for edge computing (RAMEC).

Together, these viewpoints help clarify how diverse edge deployments can be understood within a single conceptual framework, despite wide variation in requirements and implementations.

Cross-layer concerns represent system-wide requirements that span all layers and hierarchy levels. These include security, real-time capabilities, AI acceleration, virtualisation and system management. Their characteristics are not uniform but depend heavily on where computation occurs in the topology. For example, strict real-time constraints and lightweight security mechanisms may dominate at the product or deep edge, while stronger, more resource-intensive security and orchestration mechanisms are typical at higher levels, such as the cloud. This viewpoint emphasises that these concerns must be consistently addressed but adapted to the context.

The hierarchy levels describe a continuum of processing locations from the physical device to centralised cloud infrastructure. At the lowest level, the product edge resides directly on intelligent devices, followed by the deep edge, which provides slightly more computational capability close to the source of data. The gateway edge aggregates and communicates with devices, often functioning similarly to a programmable logic controller. Above this sits the network edge, then the private edge, which may represent on-premises data centres, followed by the co-location edge or cloudlet, and finally the public cloud. Each level offers increasing computational power, storage, and abstraction, but typically at the cost of higher latency and reduced immediacy of control.

The technology layers define the stack of components required to implement edge computing systems. At the foundation is the connectivity layer, which enables communication across devices and systems. Above it lies the silicon layer, representing the hardware capabilities such as CPUs, GPUs, and specialised accelerators. The operating system layer manages hardware resources and provides basic services. Middleware adds abstraction and interoperability, enabling distributed communication and orchestration. The information layer handles data models, storage, and processing, while the application layer delivers domain-specific functionality to end users or systems.

RAMEC highlights that edge computing is not a single architectural pattern but a spectrum of possible deployments. By mapping use cases across hierarchy levels, technology layers, and cross-layer concerns, it becomes clear that different scenarios impose very different requirements on both software and hardware. As a result, no universal solution exists, and systems must be carefully designed according to their specific operational context and constraints.

Since edge AI represents a rapidly emerging, highly disruptive technology driving entirely novel applications, relying solely on historical use cases for structural validation is scientifically insufficient. Consequently, the reference architecture mandates an incremental, iterative approach to practical implementation and hardware prototyping. It utilises these practical engineering steps as important alternative evidence for continuous validation and as a necessary proof of concept.

To accurately capture both the complex system construction and its dynamic usage context, the architecture establishes three primary, interdependent views: the Computing Processing Continuum View, the Technology Stack View, and the Quality Properties View [14]. These specific views strongly facilitate detailed technical decompositions, such as successfully identifying granular functional requirements or isolating specific software building blocks within broader hardware modules.

This structured decomposition allows engineers to clearly explain the intricate relationships between these elements. Furthermore, it enables the precise allocation of specific structural building blocks to critical system functions, ensuring that optimal performance, efficient resource optimisation, and highly effective exception handling are reliably realised through the complex interaction of various distributed components.

The engineering ability to systematically generate targeted, specific views is fundamental for addressing the distinct technical concerns of various project stakeholders, ranging from low-level systems developers to high-level end users.

To actively secure and maintain stakeholder support, the architecture dynamically presents complex system information in accessible formats directly relevant to specific technical interests. This capability relies entirely on the precise, standardised distinction between an architecture view and an architecture viewpoint. A view successfully expresses the complete system architecture relative to specific stakeholder concerns, while a viewpoint rigorously establishes the formalised conventions, specific model kinds, and approved analysis techniques utilised by engineers to construct and interpret that specific view.

The reference architecture serves as a comprehensive, overarching engineering framework that encompasses the rigorous architecture definition process described by ISO/IEC/IEEE 15288:2023 [10].

The RAMEC architecture contributes to the edge AI systems reference architecture by aligning with the computing processing continuum view

elements. The hierarchy levels describe the continuum of processing locations, spanning from the physical device at the edge to the centralised cloud infrastructure. By integrating this hierarchical perspective, it is possible to map and distribute computing workloads across various nodes, optimising latency, bandwidth, and resource utilisation throughout the entire network continuum.

1.6 Multi-Dimensional Edge AI Systems Reference Architecture

The proposed 3D representation of the edge AI systems reference architecture operationalises the foundational engineering principles by defining three distinct, complementary architectural views, as illustrated in Figure 1.10.

The proposed edge AI reference architecture is structured as a 3D model, providing a holistic view of the edge AI system.

The multi-view approach ensures that all aspects of the edge AI system, from physical data generators, to hardware, software, AI frameworks, and

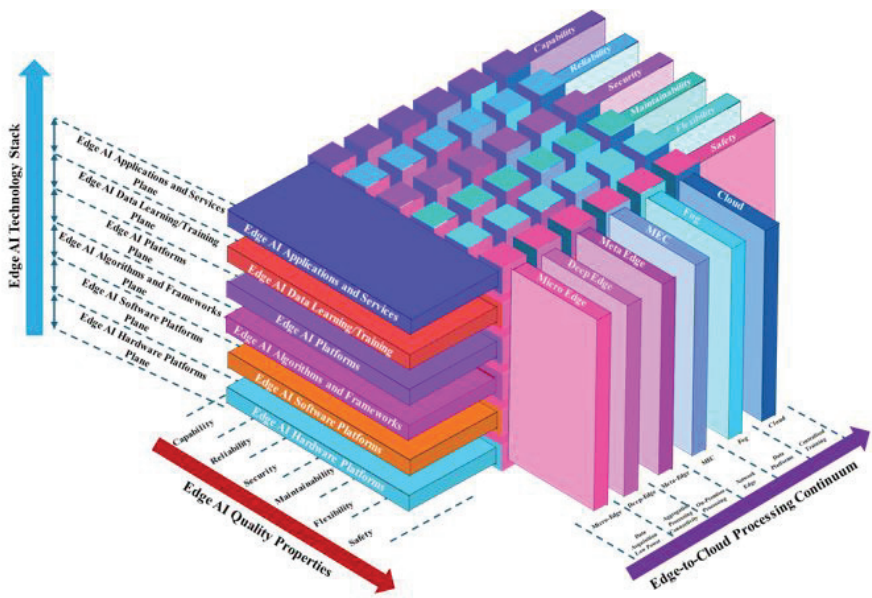

Figure 1.10 Graphical representation of the 3D edge AI reference architecture.

data layered planes, edge AI quality properties and the edge granularity across the edge-to-cloud processing continuum are addressed simultaneously.

1.6.1 Quality Properties View: Engineering for Trustworthiness and Dependability

The Edge AI Quality Properties view captures the non-functional concerns that dominate edge AI systems. Grounded in ISO/IEC 25010:2023 [12], this view emphasises dependability and trustworthiness as system-wide properties rather than isolated features.

Properties such as security, reliability, explainability, transparency, and sustainability permeate every layer of the technology stack and every tier of deployment. Treating these properties as architectural planes enables systematic specification, verification, and benchmarking of edge AI systems against clearly defined quality criteria.

The Edge AI Quality Properties view captures the key non-functional requirements that inform the design and operation of edge AI systems. Grounded in ISO/IEC 25010:2023 [12] and ISO/IEC 25012:2008 [13], this view emphasises system dependability and trustworthiness as holistic, system-wide properties, rather than treating them as isolated, bolt-on software features.

Critical properties such as cybersecurity, hardware reliability, algorithmic explainability, operational transparency, and energy sustainability must actively permeate every single layer of the technology stack and every distinct tier of physical deployment. Treating these properties as intersecting architectural planes enables the systematic specification, rigorous verification, and standardised benchmarking of complex edge AI systems against clearly defined quality criteria.

1.6.2 Technology Stack View: Layered Composition Within Each Tier

The Edge AI Technology Stack view defines the layered technical composition within each deployment tier. By explicitly structuring the system from hardware foundations through software, middleware, orchestration, frameworks and platforms, AI frameworks, and data and application layers, the view enables separation of concerns while preserving overall architectural coherence and integration. In contrast, traditional cloud-centric architectures primarily apply separation of concerns to separate software concerns within a stable infrastructure.

Thus, the key differentiator between separation of concerns in edge AI systems and traditional AI systems lies in where complexity resides and what must be separated. This view provides a consistent basis for implementing heterogeneous edge AI platforms and supports the reuse of patterns, interfaces, and standards across application domains. This consistency is essential for reducing integration risk, improving portability, and enabling comparative evaluation of alternative implementations.

In the context of edge AI systems, it is key to understand the relationship between what is being optimised and how the system is structurally designed. The quad-optimisation framework defines the specific elements being optimised, treating data as one of its four foundational pillars. To map out these optimisations, the system utilises a 3D architectural representation composed of three distinct architectural views, which serve as a method for describing and visualising these four pillars. Within this 3D model, the data pillar is represented as a comprehensive "Data Learning/Training plane" embedded within the edge AI technology stack. As data interacts with different aspects of an edge AI system, the data plane intersects with all the other planes across the different architectural views, illustrating how data flows through and unifies the entire architectural design.

Traditional cloud-centric IT architectures primarily apply the principle of separation of concerns solely to separate distinct software concerns within a stable, uniform physical infrastructure. Therefore, the key engineering differentiator between the separation of concerns in modern edge AI systems and traditional centralised AI systems lies in where the intrinsic complexity resides and what physical and logical elements must be separated.

The structured stack view provides a consistent, reproducible basis for safely implementing heterogeneous edge AI systems. It actively supports the extensive reuse of structural patterns, APIs, and data standards across diverse application domains. Architectural consistency is essential for substantially reducing systems integration risk, improving software portability, and enabling rigorous comparative evaluation of alternative engineering implementations.

1.6.3 Processing Continuum View: Partitioning Intelligence Across Edge-to-Cloud Tiers

The Edge Granularity Across the Edge-to-Cloud Processing Continuum view captures the spatial and topological distribution of computing and intelligence. Edge AI systems span multiple tiers, from micro-edge devices with

severe resource constraints to deep-edge and meta-edge to cloud infrastructures with virtually unlimited capacity.

The view explicitly defines how functionality, data processing, and decision-making responsibilities are partitioned and coordinated across the continuum. It also provides the architectural context needed to analyse latency, resilience, scalability, data, and AI sovereignty, which are critical in industrial, societal, mission-critical, and safety-critical applications.

This specific view defines exactly how critical functionality, data processing workloads, and autonomous decision-making responsibilities are securely partitioned and dynamically coordinated across the entire computing continuum. It provides the architectural context engineers need to accurately analyse network latency, system resilience, dynamic scalability, and strict data sovereignty requirements. These analytical factors are critical in industrial, societal, mission-critical, and highly safety-critical applications.

The specific processing characteristics, hardware targets, and latency constraints of the various components spanning from the extreme micro-edge to the centralised cloud are detailed in Table 1.1.

Together, these three tightly integrated views, as illustrated in Figure 1.11, form a coherent, sound architectural description that actively supports the full, rigorous systems engineering lifecycle.

These views enable rigorous theoretical analysis and precise specification during early conceptual design phases. They directly guide physical implementation through highly consistent patterns and standardised interfaces, providing a reference basis for formal system verification, operational validation, physical testing, and comparative benchmarking. By aligning with international standards such as ISO/IEC/IEEE 42010 and ISO/IEC 25010, the reference architecture establishes a shared vocabulary and a rigorous quality model, thereby facilitating seamless collaboration among diverse stakeholders and enabling strict comparability across competing vendor solutions.

The edge AI systems reference architecture serves as the foundational blueprint for implementing complex edge AI systems, transforming abstract, high-level requirements into a functional structure comprising physical hardware, embedded software, specialised AI algorithms, and secure data components. It fundamentally bridges the challenging gap between theoretical conceptual design and concrete technical implementation by defining exactly how distinct components within the edge AI technology stack must interact securely across the entire processing continuum, thereby ensuring the realisation of all required systemic quality properties.

Table 1.1 Comparative Analysis of the Edge-to-Cloud Processing Continuum Tiers.

Tier	Typical Hardware	Key Accelerators	Latency Target	Primary Function
Micro-Edge	MCUs (Cortex-M), MPUs, SoCs.	TinyML, micro-NPU, NPU.	< 1 ms	Sensing, simple inference, wake-up triggers.
Deep-Edge	Gateways, Industrial PCs	Edge GPU, TPU, VPU, NPU.	2 - 5 ms	Aggregation, sensor fusion, local control
Meta-Edge	On-premises micro-servers, high performance embedded servers.	High-end GPU, FPGA.	< 10 ms	Local training, complex analytics, orchestration.
MEC	Telecom-Grade servers, rack servers.	GPUs for packet processing, large FPGAs for network function virtualization (NFV).	< 30 ms	Content caching, high-bandwidth processing, application offloading.
Fog	Industrial routers, servers.	TPUs, DSPs, large FPGAs.	< 50 ms	Bridging OT (Operational Technology) and IT. Focus on data processing before the cloud.
Cloud	Hyperscale Datacentres.	Tensor Core GPUs, ASIC.	> 100 ms	Global training, archiving, fleet management.

The edge AI systems reference architecture significantly advances beyond RAMEC and 3D IoT layered architectures by introducing a unified conceptual framework that integrates the structural elements of both the 3D IoT architecture and RAMEC. What truly sets this new architecture apart is its incorporation of diverse AI method families, specifically mapping out where ML, DL, GenAI, and agentic models can optimally execute. The architectural concept elevates the edge AI system's design by including the edge AI model lifecycle and quality properties as dedicated quality planes. The edge AI systems reference architecture innovates by addressing the continuum of computational workload and the actual intelligence that spans

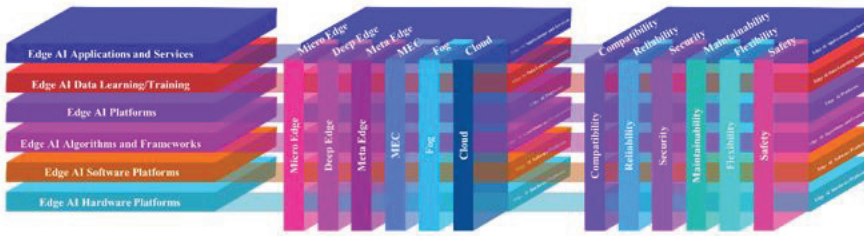

Figure 1.11 AI systems reference architecture views unfolded.

multiple architectural tiers, enabling a dynamic, distributed, and capable edge environment for edge AI systems-of-systems implementations.

Examples of complex, real-world operational scenarios anchor the practical implementation of these edge AI architectural views. Consider a fleet of autonomous delivery vehicles operating on a tiered architecture, which illustrates the edge processing granularity architectural view considering that the onboard the vehicle specialised hardware (deep-edge) runs safety-critical pedestrian detection algorithms with millisecond latency, prioritising the quality property of safety above all else by focusing on the edge AI hardware, edge AI platforms and data learning/training planes of the edge AI technology stack architectural view. At the same time, non-critical telemetry data, such as GPS coordinates of road potholes, is compressed locally and subsequently transmitted to a centralised server cluster (the cloud). This data is used to continually update and train the highly complex route optimisation software models for the entire vehicle fleet. This dual-path process demonstrates highly complex interaction across the entire processing continuum without ever compromising the individual vehicle's immediate, safety-critical reaction time.

Another example is a highly secure, remote medical facility for elderly care. Vulnerable patients constantly wear biometric smart wristbands that directly represent the extreme micro-edge computing tier. These specialised wearable devices utilise ultra-low-power microcontrollers to continuously run lightweight, optimised anomaly-detection models directly on the local silicon hardware, enabling instant identification of potentially life-threatening heart arrhythmias. By securely processing this sensitive, regulated biometric data locally rather than continuously transmitting massive raw data streams to vulnerable cloud servers, the entire system strictly adheres to stringent privacy and cybersecurity standards while simultaneously maximising the device's critical battery life. This scenario illustrates exactly how the edge

AI technology stack view must be aggressively and continuously optimised for severely resource-constrained hardware environments.

1.7 Discussion: Value, Interoperability, and Domain Maturity

The primary driver for actively adopting the edge AI systems reference architecture is the extreme heterogeneity inherent in modern multi-X edge AI environments. These complex environments comprise multiple interacting edge AI systems, utilising multiple sensory modalities, and employing multiple autonomous AI agents spanning continuously from the extreme edge to the centralised cloud.

New edge AI systems developments have fundamentally transitioned from creating simple, isolated, closed systems to engineering highly complex systems-of-systems that feature distributed intelligence.

The paradigm shift requires coordinated engineering efforts spanning multiple spatial nodes and sites, diverse solution stakeholders, and varied scientific disciplines. As the physical scope and computational complexity of edge AI systems increase, so does the immense engineering difficulty of safely maintaining structural coherence and operational stability across these distributed systems.

Edge AI technologies are rapidly maturing, and a formalised reference architecture for edge AI systems is required as the sheer multiplicity of competing edge solutions reaches a critical mass in the global market. Without a shared engineering framework, successfully integrating diverse edge AI systems developed across completely different applications and distinct industries becomes massively inefficient, highly expensive, and inherently error prone.

The edge AI systems reference architecture addresses this critical bottleneck by providing a common technical lexicon and a standardised functional taxonomy. This standardisation enables diverse engineering teams to communicate precisely and effectively, and to align their development efforts toward a shared, secure edge AI architectural vision.

The widespread industrial implementation of an edge AI systems reference architecture delivers increased economic and technological value by driving and harvesting synergy across the entire edge AI domain. It allows architects to identify exactly where shared software assets and hardware standardisation can be effectively and profitably applied, and conversely, where such standardisation might be technologically counter-productive. This

deep strategic insight enables rapid, efficient development of robust edge AI solutions, standardised edge AI workflows, product lines, and technological portfolios. The approach reduces the significant time and financial costs typically associated with continually reinventing complex technical solutions for problems that have already been solved in engineering.

The edge AI systems reference architecture can improve interoperability among evolving, disparate edge AI systems. By explicitly and properly modelling critical system functions and operational qualities above the localised single-system level, engineers can ensure strict backward compatibility and facilitate smoother, safer over-the-air system upgrades for edge AI-driven systems. The approach directly leads to reduced systems integration costs and improved overall dependability of deployed edge AI systems.

The edge AI systems reference architecture serves as a foundational, baseline, shared starting point that firmly anchors future technical discussions and architectural changes, thereby mitigating the risks typically associated with the evolution and deployment of complex edge AI systems.

The collaboration across the European edge AI ecosystem brought together complementary expertise from multiple projects, which significantly increased the value of the reference architecture. Instead of addressing isolated challenges, the joint effort aligned advances in hardware, software, toolchains, and applications. This created a more coherent framework that reflects real-world system complexity, making the architecture not just theoretical but directly usable by industry and research communities.

Using the edge AI systems reference architecture, interoperability can be improved by addressing technical, syntactic, and semantic interoperability issues across the stack, from low-level AI chips to high-level applications. By sharing requirements and solutions, the reference architecture supports defining common interfaces, data flows, and integration patterns. For example, a smart manufacturing system defined using the edge AI systems reference architecture can combine edge devices from different vendors with AI models developed in separate toolchains while still adhering to a unified architectural approach.

The ecosystem collaboration and cooperation accelerated the maturity of the edge AI domain. By validating ideas across multiple projects and use cases, the architecture reflects tested practices rather than isolated concepts. In energy systems, for instance, edge AI can monitor and optimise distributed energy systems and lighting luminaries with intelligent capabilities in real time, while in mobility, it supports autonomous mobile systems, software-defined vehicles (SDV), and AI-defined vehicles (AIDV) functions at the

edge. These diverse validations strengthen confidence in the architecture and make it applicable across sectors.

The reference architecture is particularly valuable because it adapts to the needs of different industries.

In agrifood, it enables precision farming by combining sensor data and local AI inference. In healthcare, it supports wearable devices that process sensitive data locally to preserve privacy. In digital society applications, such as smart cities, it helps coordinate distributed intelligence across cameras, sensors, and infrastructure.

Each sector benefits from a common structure while tailoring specific components to its requirements.

Looking ahead, the architecture provides a foundation for emerging edge AI trends by offering a holistic view of how technologies fit together. It is designed to accommodate embodied AI systems, such as robots that interact with the physical world, as well as generative AI models running at the edge for real-time content creation.

The architecture also supports agentic AI, where autonomous agents make decisions locally, and immersive technologies such as augmented (AR), virtual (VR), mixed (MR), and extended (XR) reality that require low-latency intelligence close to users and integration into metaverse, omniverse and multiverse simulation platforms.

In a future smart factory, the reference architecture can support the implementation of edge AI systems-of-systems where embodied AI robots could collaborate with human workers, generative AI could assist in design and troubleshooting directly on-site, and agentic systems could autonomously manage workflows.

The reference architecture helps ensure that all these elements interoperate in real-time, providing a shared blueprint for building complex, distributed, and intelligent edge AI systems-of-systems.

Acknowledgements

This publication has received funding through the projects Chips JU EdgeAI and HE dAIEDGE. The Chips JU EdgeAI “Edge AI Technologies for Optimised Performance Embedded Processing” project is supported by the Chips Joint Undertaking and its members, including top-up funding by Austria, Belgium, France, Greece, Italy, Latvia, the Netherlands, and Norway under grant agreement No 101097300. The HE dAIEDGE “A network of excellence

for distributed, trustworthy, efficient and scalable AI at the Edge" project is supported under grant agreement No 101120726. Funded by the European Union. Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or the Chips Joint Undertaking. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] J. Dietz, "Architecture – Building strategy into design." Netherlands Architecture Forum. Academic Service – SDU, The Hague (2008), http://www.naf.nl ISBN-13: 9789012580861.

[2] D. Greefhorst and E. Proper "Architecture Principles - The Cornerstones of Enterprise Architecture." Springer-Verlag Berlin Heidelberg 2011. ISBN 978-3-642-20278-0, e-ISBN 978-3-642-20279-7, doi: 10.1007/978-3-642-20279-7. https://sar.ac.id/stmik_ebook/prog_file_file/f7KjsxWOBJ.pdf.

[3] F. Harmsen, H. Proper, and N. Kok, "Informed Governance of Enterprise Transformations." In: Proper HA, Harmsen AF, Dietz JLG (eds) Advances in enterprise engineering II—Proceedings of the first NAF academy working conference on practice-driven research on enterprise transformations, PRET 2009. Held at CAiSE 2009, Amsterdam, The Netherlands, June 2009. Lecture notes in business information processing, vol 28. Springer, Berlin, pp 155–180. ISBN-13:9783642018589. https://www.erikproper.eu/publications/EP-2024-05-22-12-14-32.pdf.

[4] Industry IoT Consortium®(IIC®). The Industrial Internet Reference Architecture. Version 1.10. https://www.iiconsortium.org/wp-content/uploads/sites/2/2022/11/IIRA-v1.10.pdf.

[5] ISO/IEC/IEEE 42010:2022. Software, systems and enterprise - Architecture description https://www.iso.org/standard/74393.html.

[6] ISO/IEC/IEEE 42020:2019. Software, systems and enterprise - Architecture processes. https://www.iso.org/standard/68982.html.

[7] ISO/IEC/IEEE DIS 42024. Enterprise, systems and software - Architecture fundamentals. https://www.iso.org/standard/87510.html.

[8] ISO/IEC/IEEE 42030:2019. Software, systems and enterprise - Architecture evaluation framework. https://www.iso.org/standard/73436.html.

[9] ISO/IEC/IEEE DIS 42042. Enterprise, systems and software — Reference architectures. https://www.iso.org/standard/87310.html.
[10] ISO/IEC/IEEE 15288:2023. Systems and software engineering - System life cycle processes. https://www.iso.org/standard/81702.html.
[11] ISO 15704:2019. Enterprise modelling and architecture - Requirements for enterprise-referencing architectures and methodologies. https://www.iso.org/standard/71890.html.
[12] ISO/IEC 25010:2023. Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Product quality model. https://www.iso.org/standard/78176.html.
[13] ISO/IEC 25012:2008. Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Data quality model. https://www.iso.org/standard/35736.html.
[14] Chips JU EdgeAI Project. Deliverable D5.07. Edge AI systems reference architecture. 2026.
[15] O. Vermesan et al., "The Next Generation Internet of Things - Hyperconnectivity and Embedded Intelligence at the Edge." In, O. Vermesan and J Bacquet, "Next Generation Internet of Things Distributed Intelligence at the Edge and Human Machine-to-Machine Cooperation". 2018. https://www.riverpublishers.com/pdf/ebook/chapter/RP_9788770220071C3.pdf.
[16] O. Vermesan, "Edge AI Reference Architecture", European Conference on EDGE AI Technologies and Applications – EEAI 2025, 22 October 2025 Naples, Italy. https://doi.org/10.5281/zenodo.17666483.
[17] O. Vermesan, "The Edge AI Systems Reference Architecture - Orchestrating Autonomous and AI-Defined Systems Through GenAI and Agentic AI in the Intelligence Mesh", HiPEAC Conference 27 January 2026, Kraków, Poland. https://doi.org/10.5281/zenodo.18816484.
[18] O. Vermesan, The strategic imperative of Edge AI systems reference architecture, INSIDE Magazine Issue 12, March 2026, pp.18-23.
[19] US Department of Defense, Office of the Assistant Secretary of Defense, Networks and Information Integration (OASD/NII), Reference Architecture Description, June 2010. https://dodcio.defense.gov/Portals/0/Documents/Ref_Archi_Description_Final_v1_18Jun10.pdf.
[20] The Open Group Architecture Framework (TOGAF) Version 10, https://www.opengroup.org/togaf.
[21] The Open Group. The TOGAF®10^{th} Edition Standard. Introduction and Core Concepts. 2022. Van Haren Publishing, 's-Hertogenbosch -

NL, SBN eBook: 978 94 018 0860 6. https://www.avtechcn.com/pdf/togaf10part01.pdf.

[22] Willner and V. Gowtham, "Toward a Reference Architecture Model for Industrial Edge Computing," in *IEEE Communications Standards Magazine*, vol. 4, no. 4, pp. 42-48, December 2020, https://www.doi.org/10.1109/MCOMSTD.001.2000007.