INTERNATIONAL STANDARD

ISO/IEC/IEEE 42010

Second edition
2022-11

Software, systems and enterprise — Architecture description

Logiciel, systèmes et entreprise — Description de l'architecture

Reference number
ISO/IEC/IEEE 42010:2022(E)

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO/IEC documents should be noted. This document was drafted in accordance with the rules given in the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see https://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

ISO/IEC/IEEE 42010 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*, in cooperation with the Software and Systems Engineering Standards Committee of the Computer Society of the IEEE, under the Partner Standards Development Organization cooperation agreement between ISO and IEEE.

This second edition cancels and replaces the first edition (ISO/IEC/IEEE 42010:2011), which has been technically revised.

The main changes are as follows:

— The term used to refer to the subject of an architecture description is changed from “system of interest” to “entity of interest” (EoI) to be compatible with ISO/IEC/IEEE 42020 and ISO/IEC/IEEE 42030 standards and to allow for its application in non-system architecture situations. The term “entity” is also used in this document when entities are considered as surrounding things in an environment of an EoI.

— The term “architecture description framework” (ADF) replaces “architecture framework” in the previous edition. It is defined in order to differentiate ADFs from other kinds of architecting frameworks like architecture evaluation frameworks specified in ISO/IEC/IEEE 42030.

— Architecture description element, introduced in the 2011 edition (see ISO/IEC/IEEE 42010:2011, 4.2.6, 5.7 and A.6) is now defined in Clause 3 as identified or named part of an architecture description allowing representing at least stakeholders, concerns, perspectives, and aspects identified in an AD, and views, view components, viewpoints, and model kinds included in an AD.

— Aspect and stakeholder perspective concepts —already introduced in the 2011 edition (See 3.5, note 1 of 5.6, Annex A and B) are defined and described to accommodate current practice where these ideas are prevalent.

— A correspondence defines an identified or named relation between AD elements, as in Clause 4.2.6 of the 2011 edition. But, to clarify the relationship between AD and correspondence, a note 1 to the definition is added to state that for the purpose of correspondences, an architecture description can be considered as an AD element in another architecture description. This correspondence between ADs is necessary because an architecture can be described by more than one AD and these alternatives of architectures have related for activities like trade-off analysis and decision making.

— The term “architecture view component” is introduced as a separable portion of one or more architecture views, replacing “architecture model” in the 2011 edition. This change is to account for the fact that some parts of a view are model-based while others may not be. View components can be derived from an information source, which can sometimes be a model.

— Model-based view components are governed by model kinds and documented by legends. Non-model-based view components are documented by legends.

— Model kinds are identified as a new conformance case to encourage model-based architecting.

— The concept of architecture viewpoint is updated to accommodate current practice where a viewpoint governs one or more architecture views within an AD.

— The definition of “model kind” given by the 2011 edition is extended to include categories of models as used by ADF like UAF.

— The figures use an informal entity-relationship diagram notation replacing UML class diagrams in the 2011 edition, to facilitate comprehension by users of this document. The multiplicities of the relationships are explained in the text when necessary.

— Annex E illustrates a few concepts pertaining to architecture life cycles and architecture description life cycles.

— Annex F shows examples of how some architecture description frameworks can conform to requirements of this document.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

The complexity of human-made entities has grown to an unprecedented level. This has led to new opportunities, and also increased challenges for organizations that create and use these entities. Architecting is increasingly applied by organizations, teams and individuals, to help manage the complexity faced by stakeholders of these entities.

Examples of entities include the following: Enterprise, organization, solution, system (including software systems), subsystem, process, business, data (as a data item or data structure), application, information technology (as a collection), mission, product, service, software item, hardware item, product line, family of systems, system of systems, collection of systems, collection of applications.

An architecture of an entity, expressed in one or more architecture descriptions (AD), assists in understanding the fundamental concepts or properties of the entity, pertaining to its structure, behaviour, design and evolution, such as feasibility, utility and maintainability and fundamental concepts for its development, operation, employment, external impacts, utilization and decommissioning.

ADs are used by the parties that create, use and manage human-made entities to improve communication and cooperation, enabling all parties, organizations, teams and individuals to work together in an integrated and coherent fashion.

NOTE ISO/IEC/IEEE 42020 specifies a set of processes for architecting which can be employed in support of creating one or more ADs. The architecture elaboration process in ISO/IEC/IEEE 42020 is especially relevant for creation of ADs.

Whereas an AD is a tangible work product, an architecture is intangible and abstract, understood through its concepts, properties and principles.

Architecture description frameworks (ADF) are used to codify the conventions and common practices of architecture description. Architecture description languages (ADL) are used to codify the description of architectures within different communities and domains of application.

ADs have many uses, such as design, development, documentation, analysis, evaluation, maintenance, risk mitigation, downstream user specifications, tool specification, communication, planning, guidance, life cycle support, decision support, review, training, design validation, solution trade studies, cost comparison and analysis, by a variety of stakeholders throughout the life cycles of their entities of interest. Annex D describes more uses of an AD.

This document provides terms, definitions and relationships for best practices in ADs. The provisions of this document serve to specify desired properties of ADs. This document also gives provisions that specify desired properties of ADFs and ADLs in order to usefully support the development and use of ADs. This document provides a basis for considering and comparing ADFs and ADLs by providing a common ontology for specifying their contents.

This document can be used to establish a coherent architecting practice for developing ADs, ADFs and ADLs within an organization, in the context of an entity of interest (EoI) or its architecture. The provisions of this document can be used to assess conformance of specifications of ADs, ADFs, ADLs, viewpoints and model kinds.

The intent of this document is to enable a range of consistent and coherent approaches to describing an architecture including document-centric and model-based techniques.

This document also provides motivations for use of architecture-related terms and concepts in other documents such as guides and standards.

Users of this document are advised to consult Clause 5 to gain appreciation of the conceptual foundations, along with the concepts and principles associated with an AD work product.

This document does not explicitly address completeness or correctness regarding the inclusion of particular elements in an AD. Nevertheless, completeness and correctness of an AD can be partially checked, for example, through the consistency of the AD elements established, whether relationships are transitive, and whether AD elements are shown in the views. Consistency rules can also be defined by showing whether the same particular AD element has correspondences with an AD. In addition, specifications that appear as elements within an AD are expected to be complete, precise and verifiable with respect to the subject of the specification.

In this document, the following verbal forms are used:

— "shall" indicates a requirement;

— "should" indicates a recommendation;

— "may" indicates a permission.

Software, systems and enterprise — Architecture description

1 Scope

This document 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.

This document distinguishes the architecture of an entity of interest from an AD expressing that architecture. Architectures are not the subject of this document.

This document specifies requirements for use of the architectural concepts and their relationships as captured in an AD. It does not specify requirements for any entity of interest or its environment.

This document specifies requirements for an architecture description framework (ADF), an architecture description language (ADL), architecture viewpoints and model kinds in order to usefully support the development and use of an AD.

This document specifies conformance to the requirements for an AD, ADF, ADL, architecture viewpoint and model kind.

This document does not specify the processes, architecting methods, models, notations, techniques or tools by which an AD is created, utilized or managed.

This document does not specify any format or media for recording an AD.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO, IEC and IEEE maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp/ui

— IEC Electropedia: available at https://www.electropedia.org/

— IEEE Standards Dictionary Online: available at https://dictionary.ieee.org/

NOTE For additional terms and definitions in the field of systems and software engineering, see ISO/IEC/IEEE 24765, which is published periodically as a “snapshot” of the SEVOCAB (Systems and software Engineering Vocabulary) database and is publicly accessible at www.computer.org/sevocab.

3.1
architecting
conceiving, defining, expressing, documenting, communicating, certifying proper implementation of, maintaining and improving an *architecture* (3.2) throughout the life cycle of an *entity of interest* (3.12)

3.2
architecture
fundamental concepts or properties of an entity in its *environment* (3.13) and governing principles for the realization and evolution of this entity and its related life cycle processes

[SOURCE: ISO/IEC/IEEE 42020:2019, 3.3, modified — The notes to entry have been removed.]

3.3
architecture description
AD
work product used to express an *architecture* (3.2)

Note 1 to entry: A work product is an artifact produced by a process (see ISO/IEC 20246:2017, 3.18).

Note 2 to entry: An AD is a tangible representation of information provided to the *stakeholders* (3.17). An AD is considered an *information part* (3.14).

3.4
architecture description element
AD element
identified or named part of an *architecture description* (3.3)

Note 1 to entry: AD elements include *stakeholders* (3.17), *concerns* (3.10), *stakeholder perspectives* (3.18), and *aspects* (3.9) identified in an *AD* (3.3), *ADLs* (3.6), *ADFs* (3.5) and *correspondences* (3.11) and correspondence methods used in an AD, and *architecture views* (3.7), *view components* (3.19), *architecture viewpoints* (3.8), and *model kinds* (3.15) included in an *AD* (3.3).

Note 2 to entry: For the purpose of *correspondences* (3.11), an *AD* (3.3) can be considered as an AD element in another *AD* (3.3).

3.5
architecture description framework
ADF

conventions, principles and practices for the description of *architectures* (3.2) established within a specific domain of application or community of *stakeholders* (3.17)

EXAMPLE Generalized Enterprise-Referencing Architectures Modelling Framework (GERAM) (ISO 15704:2019, Annex B), Reference Model of Open Distributed Processing (RM-ODP),[2] Unified Architecture Framework (UAF)[48], and NATO Architecture Framework (NAF)[44].

Note 1 to entry: Architecture description frameworks promote structured organization, consistency of description, greater potential for reuse, and completeness of *architecture views* (3.7) and models.

3.6
architecture description language
ADL
means of expression, with syntax and semantics, consisting of a set of representations, conventions, and associated rules intended to be used to describe an *architecture* (3.2)

EXAMPLE Architecture Analysis and Design Language (AADL),[57] ArchiMate,[61] UML,[49] SysML,[47] UAF Profile[48].

3.7
architecture view
information part (3.14) comprising portion of an *architecture description* (3.3)

EXAMPLE An Information or Data View addresses information-relevant concerns framed by an Information viewpoint. It contains as *view components* (3.19), a conceptual data model, a data management model and a data access model and correspondences linking those components together.

3.8
architecture viewpoint
set of conventions for the creation, interpretation and use of an *architecture view* (3.7) to frame one or more *concerns* (3.10)

Note 1 to entry: In this document, "to frame" concerns means "to shape, compose, give expression to" those concerns. It is used to distinguish the stages of framing concerns by a viewpoint from addressing those concerns in a resulting view. This is analogous to the distinction between "framing a problem" and "solving that problem".

Note 2 to entry: A viewpoint is a frame of reference for the concerns determined by the architect as relevant to the purpose of the *architecture description* (3.3).

Note 3 to entry: The conventions of an architecture viewpoint are documented in a *specification* (3.16) of that viewpoint. In some communities and architecture description frameworks, "view specification" and viewpoint are synonyms.

Note 4 to entry: The identification of a viewpoint is often the result of prior knowledge, experience and praxis in the domain(s) to which the viewpoint applies, indicating the information relevant to addressing the *concern* (3.10).

3.9
aspect
part of an entity's character or nature

EXAMPLE Functional, structural and informational aspects of an entity.

Note 1 to entry: A particular aspect can be used for capturing the relevant features of the *entity of interest* (3.12) as a refinement of one or more *concerns* (3.10) under examination with respect to some part of its character, e.g. the structural character, functional character or informational character of the entity.

Note 2 to entry: Aspects enable the architect to analyse, address and structure *concerns* (3.10). In general, there is a many-to-many relation between aspects and *concerns* (3.10).

Note 3 to entry: See 5.2.5 for more discussion and examples.

3.10
concern
matter of relevance or importance to a *stakeholder* (3.17)

Note 1 to entry: concerns can be identified with regards to an *entity of interest* (3.12) or independently, such as with regards to environment, scenario, situation or use case of that entity.

Note 2 to entry: In this document, interest in an entity is intended to encompass interest in that entity's *environment* (3.13), life cycle, *architecture* (3.2), requirements, design, implementation and operation. Such interests are captured via *aspects* (3.9), concerns and *stakeholder perspectives* (3.18).

Note 3 to entry: The identification of a concern is often the result of prior knowledge, experience and praxis in the domain to which the concern applies.

Note 4 to entry: See 5.2.3 for more discussion and examples.

[SOURCE: ISO/IEC/IEEE 42020:2019, 3.8, Notes have been modified]

3.11
correspondence
identified or named relationship between two or more *architecture description elements* (3.4)

EXAMPLE Correspondences are used to express a wide range of relationships, such as equivalence, composition, refinement, consistency, traceability, dependency, constraint, satisfaction, and obligation.

Note 1 to entry: For the purpose of correspondences, an *architecture description* (3.3) can be considered as an *AD element* (3.4) in another *architecture description* (3.3).

Note 2 to entry: Correspondences can be identified or named relationship between *architecture description elements* (3.4) in different *architecture descriptions* (3.3) or between *architecture description elements* (3.4) stated in different notations.

3.12
entity of interest
EoI
subject of an *architecture description* (3.3)

EXAMPLE Enterprise, organization, solution, system (including software systems), subsystem, process, business, data (as a data item or data structure), application, information technology (as a collection), mission, product, service, software item, hardware item, product line, family of systems, system of systems, collection of systems, collection of applications.

Note 1 to entry: In this document, the term entity of interest refers to the entity whose *architecture* (3.2) is under consideration in the preparation of an *architecture description* (3.3).

Note 2 to entry: This document distinguishes the entity of interest from other entities which are not the subject of the *architecture description* (3.3).

Note 3 to entry: In this document, interest in an entity is intended to encompass interest in that entity's environment, life cycle, architecture, requirements, design, implementation and operation. Such interests are captured via aspects, concerns and stakeholder perspectives.

3.13
environment
context of surrounding things, conditions, or influences upon an entity

Note 1 to entry: The environment of an *entity of interest* (3.12) includes external entities that can have various influences upon an *entity*, such as developmental, technological, business, operational, organizational, political, economic, legal, regulatory, ecological and social influences as well as external physical effects such as electromagnetic radiation, charged particles, gravitational effects, and electric and magnetic fields.

Note 2 to entry: A label attached as a qualifier to the term environment identifies a particular context within another context, such as development environment, test environment, and operational environment.

3.14
information part
separately identifiable body of information that is produced, stored, and delivered for human and machine use

3.15
model kind
category of model distinguished by its key characteristics and modelling conventions

EXAMPLE Functional models, activity models, structural models, use case models, geopolitical models, analytic models and economic models.

3.16
specification
information part (3.14) that identifies, in a complete, precise and verifiable manner, the requirements, design, behaviour, or other expected characteristics of an entity

[SOURCE: ISO/IEC/IEEE 15289:2019, 3.1.26 — "a system, service or process" replaced with "an entity", "information item" replaced with "information part"]

3.17
stakeholder
role, position, individual, organization, or classes thereof, having an interest, right, share, or claim, in an *entity of interest* (3.12)

EXAMPLE End users, operators, acquirers, owners, suppliers, architects, developers, builders, maintainers, regulators, taxpayers, certifying agencies, and markets.

3.18
stakeholder perspective
way of thinking about an *entity of interest* (3.12), especially as it relates to *concerns* (3.10)

EXAMPLE The labels given to the middle three rows (i.e. owner, designer and builder) of the Zachman framework[67] correspond to stakeholder perspectives. The rows in the Unified Architecture Framework[48] and NATO Architecture Framework[44] grids correspond to stakeholder perspectives (although they are called ”domains” and “subjects of concerns,” respectively in those frameworks). See 5.2.4 for more examples.

Note 1 to entry: The way one thinks about an entity can be influenced by one’s beliefs, training, experience, knowledge, personality, character traits, culture, peer pressure, role or stance, etc.

3.19
view component
architecture view component
separable portion of one or more *architecture views* (3.7) that is governed by the applicable *model kind* (3.15) or legend

EXAMPLE An architecture view component describing access control mechanisms can be used in several views of an *architecture description* (3.3) to explain functional flows, behaviour and security features of an entity.

Note 1 to entry: In the context of an *architecture description* (3.3), a legend is an informal documentation of conventions.

4 Conformance

The requirements in this document are contained in Clauses 6, 7 and 8. There are five situations in which claims of conformance with the provisions of this document can be made.

1) When conformance is claimed for an architecture description, the claim shall demonstrate that the specification of the architecture description meets the requirements listed in Clause 6.

2) When conformance is claimed for an architecture description framework, the claim shall demonstrate that the specification of the architecture description framework meets the requirements listed in 7.1.

3) When conformance is claimed for an architecture description language, the claim shall demonstrate that the specification of the architecture description language meets the requirements listed in 7.2.

4) When conformance is claimed for an architecture viewpoint, the claim shall demonstrate that the specification of the architecture viewpoint meets the requirements listed in 8.1.

5) When conformance is claimed for a model kind, the claim shall demonstrate that the specification of the model kind meets the requirements listed in 8.2.

This document is designed such that “tailoring” is neither required nor permitted for its use when claims of conformance are made.

5 Conceptual foundations

5.1 General

This clause introduces the conceptual foundations of architecture description expressed in a set of conceptual models (see 5.2) and the application of those foundations to ADs (5.2), ADFs (see 5.4.2) and ADLs (see 5.4.3). The use of the architecture descriptions to support different architecture practices is outlined in Annex D. The concepts introduced in this clause are used in Clauses 6 to 8 to express requirements.

NOTE Annex A provides further discussion of the terms and concepts used in this document and presents examples of their use in an historical context.

5.2 Conceptual models of an architecture description

5.2.1 Context of architecture description

The term "entity of interest" is used in this document to refer to the subject of an architecture description. The term is intended to encompass, but is not limited to, entities within the following fields of application, reflecting the intended scope of this document as specified in Clause 1.

— software, including software products and services, per ISO/IEC/IEEE 12207;

— systems, including one-of-a-kind systems, mass-produced systems, customized, adaptive systems, stand-alone and embedded systems, per ISO/IEC/IEEE 15288;

— enterprises as described in ISO 15704, i.e. human undertakings or ventures that have mission, goals and objectives to offer products or services, or to achieve a desired project outcome or business outcome.

This document takes no position on what constitutes an entity within those or other fields of application or elsewhere. An entity can be a concrete entity or an abstract entity. An AD as specified in this document is suitable not only for entities in the fields of applications listed above, but also for entities in fields such as natural systems or conceptual systems.

Each entity of interest is situated in an environment which influences its characteristics and behaviours. The environment determines the totality of influences upon the entity of interest and the totality of influences of the entity of interest upon that environment, including its interactions with the environment and other entities, throughout the life cycle of that entity of interest.

Figure 1 depicts key concepts pertaining to an entity of interest and its architectures as a means of understanding ADs.

NOTE 1 The figures and text in the remainder of Clause 5 constitute a set of conceptual models of architecture description. Figures 1 to 6 use an informal entity-relationship diagram notation to facilitate comprehension by users of this document. In the figures, rounded rectangles represent information objects, and arrows represent relationships between objects with the annotation read in the arrow direction. The figures illustrate the key concepts described throughout Clause 5. Annex A presents the full conceptual model.

NOTE 2 Identification of the EoI can emerge from the analysis of the concerns of the stakeholders or can preexist before identification of some stakeholders and their concerns.

EXAMPLE The identification of an EoI generally results from the definition of the problem space. The problem description can be expressed with an architecture definition of a set of operational capabilities (often called “capability architecture”). At this capability definition stage, identified stakeholders are potentially concerned by the future EoI.

5.2.2 Architectures and architecture descriptions

The architecture of an entity of interest comprises the fundamental concepts or properties of that entity considered in its environment. The architecture of an entity of interest can pertain to any or all of the entity’s:

— constituent elements;

— interactions or interrelationships among its elements;

— interactions or interrelationships with its environment, including with other entities in that environment;

— behaviour and structure;

— principles governing its design, use, operation and evolution.

An AD is an expression of an architecture. ADs are work products resulting from architecting efforts. As a work product, an AD is devised for the specific purpose for which the architecting effort is undertaken, which is distinct from the purpose of the entity of interest. An AD comprises AD elements (see 5.2.10).

The architecture of an entity of interest can be understood through one or more distinct ADs, each created for a purpose relative to the architecture and stakeholder needs. Different ADs can, for example, be based on different stakeholders (see 5.2.3), stakeholder perspectives (see 5.2.4), time periods (sometimes termed epochs), or specific contexts or usage within the environment.

NOTE ISO/IEC/IEEE 42020 specifies a set of processes for architecting which can be employed in support of creating one or more ADs.

5.2.3 Stakeholders and concerns

Stakeholders are parties with direct or indirect interests in an entity. Among the stakeholders are those parties that have influence or control over and those who are impacted by an entity. A stakeholder's interests are typically expressed as concerns about an entity of interest or the architecture of which they are aware. Concerns are often the result of the stakeholder's perspective gained from domain knowledge, experience, training, responsibility and authority.

Concerns are matters of interest or importance to one or more stakeholders. A concern can be shared by one or more stakeholders and a stakeholder can hold more than one concern. The legitimacy and importance of a concern held by a stakeholder can be a consequence of the role of the stakeholder (e.g. owner, end user or participant, developer, architect, maintainer, disposer) or financial or social rights, shares, impact or claims (e.g. funding organization, governmental body, party receiving environmental impact from entity, stakeholder of an entity impacted by the entity of interest).

Some stakeholders' concerns are contrary to the success of the entity of interest. These stakeholders can have disagreements on the grounds of political or environmental considerations, can seek active disruption of the entity's operations, or even outright destruction of the entity. Adversarial concerns can be taken into account when developing the architecture of the entity. For example, political objections can be resolved by incorporating a negotiated solution in the architecture of the entity, or threats can be mitigated by taking preventative measures.

During the entity of interest's life cycle, concerns can arise at any time including (but not limited to) during conceptualization, when design choices are made, from construction or implementation, through deployment, operation, transfer of ownership, retirement and disposal.

Concerns can manifest in various ways in relation to stakeholder's needs, architecture goals, expectations, responsibilities, requirements, design constraints and assumptions. Concerns can also manifest in recognition of dependencies, quality attributes, architecture decisions, risks or other issues.

Concerns can pertain to influences exerted upon or by an entity of interest, including developmental, technological, business, operational, organizational, political, economic, legal, regulatory, ecological, social and physical influences. Concerns can also pertain to design influences such as internal structural features and component interoperability, particularly when architecting a system of systems or an enterprise.

EXAMPLES

— How is the system maintained?

— What system behaviours are safety-critical?

— Can the entity of interest attain compliance with national regulations?

— What is the cost to operate?

— What are the risks, opportunities, satisfaction, resilience, coherence, affordability, complexity and trust offered by this architecture?

— What are the distribution transparencies described in the Reference Model of Open Distributed Processing[1]?