IEC SRD 63188

Edition 1.0 2022-09

SYSTEMS REFERENCE DELIVERABLE

colour inside

Smart cities reference architecture methodology

IEC Secretariat
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

IEC SRD 63188

Edition 1.0 2022-09

SYSTEMS REFERENCE DELIVERABLE

colour inside

Smart cities reference architecture methodology

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 03.100.70; 13.020.20

ISBN 978-2-8322-4569-9

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

SMART CITIES REFERENCE ARCHITECTURE METHODOLOGY

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.

2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.

3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.

5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.

6) All users should ensure that they have the latest edition of this publication.

7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.

8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

IEC SRD 63188, which is a Systems Reference Deliverable, has been prepared by IEC systems committee Smart Cities: Electrotechnical aspects of Smart Cities.

The text of this Systems Reference Deliverable is based on the following documents:

Draft	Report on voting
SyCSmartCities/229/DTS	SyCSmartCities/255/RVDTS

Full information on the voting for the approval of this systems reference document can be found in the report on voting indicated in the above table.

The language used for the development of this Systems Reference Deliverable is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/standardsdev/publications.

In this document, the following print types are used.

- In 5.3, 5.4, 5.5, 5.14, 5.16 and 5.17, bold is used for text fragments that will be used in some dependency matrix.
- In 8.7.2, 8.9.2, 8.10.2 and 8.11.2, various colours are used to link interrogatives and placeholders <> within questions.

This document contains attached files in the form of Excel spreadsheets. These files are intended to be used as a complement and do not form an integral part of the document. Three files are attached:

- "SCRAM Figures 1-8, 23 and 24.xlsm";
- "SCRAM Figure 26.xlsm";
- "SCRAM Figure 28.xlsm".

All these files contain some macros for generation of illustrations. However, these macros are not necessary for viewing illustrations.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The "colour inside" logo on the cover page of this document indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

The Smart Cities Reference Architecture Methodology (SCRAM) is an adaptation of the summary of the IEC Systems Resource Group (SRG) work on systems approach (see Annex B) for the smart cities system domain. The purpose of the SCRAM is to provide a common methodology for developing the Smart Cities Reference Architecture (SCRA) which will be used as a common and tailorable template for architectures of, practically, any city system.

In 2017, when IEC SyC Smart Cities decided to work on the SCRAM, it was known that there were already various reference architectures for smart cities (see Annex C). However, the majority of these reference architectures were created under different and not publicly agreed methodologies, thus it is very difficult to make them work together. The scope of IEC SyC Smart Cities required the development of a reference architecture which is widely understood and widely used. Therefore, IEC SyC Smart Cities needed to develop the SCRAM as a publicly agreed way of doing this.

Although the primary responsibility of IEC SyC Smart Cities is electrotechnical aspects of smart cities, the SCRAM (and the SCRA) covers city systems as a whole because the electrotechnical aspects can only be properly understood in this wider context. The standardization of system elements defined by the SCRA will be done, depending on the nature of those system elements, within various SDOs Thus, the SCRAM and the SCRA are not limited to electrotechnical aspects only.

The SCRAM provides a methodology for the SCRA to achieve necessary granularity of city system elements to allow the development of practical standards. This is governed by the scope of IEC SyC Smart Cities, which is defined by the IEC SMB as "To foster the development of standards in the field of electrotechnology to help with the integration, interoperability and effectiveness of city systems."

The SCRA covers stages 1 to 4 of summary of the SRG work on systems approach – domain analysis, system architecting, use case analysis and system modelling. Effectively, the SCRA will define for smart cities a set of capabilities, data structures, processes and interfaces which can be standardized to simplify implementation of various smart cities. This set will be used as an input for the stages 5 and 6 of the summary of the SRG work on systems approach – standards analysis and gap analysis – to map the existing standards to this set and to identify standardization opportunities to existing SyCs, TCs/SCs and other SDOs.

SMART CITIES REFERENCE ARCHITECTURE METHODOLOGY

1 Scope

This document, which is a Systems Reference Deliverable, proposes a methodology that defines a coherent structure for developing Smart Cities Reference Architecture (SCRA). The Smart Cities Reference Architecture Methodology (SCRAM) reviews and defines the desired characteristics of Smart Cities, diverse SCRA viewpoints and corresponding SCRA model types in order to promote consistency and uniformity across architectures for various smart cities.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC/IEEE 42010:2011, *Systems and software engineering – Architecture description*

3 Terms and definitions

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

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

- IEC Electropedia: available at https://www.electropedia.org/
- ISO Online browsing platform: available at https://www.iso.org/obp

NOTE Additional information about some terms can be found in Annex B.

3.1 Information technology

3.1.1
data
symbols and signals such as numbers, words or other signs that represent discrete facts about an objective reality

Note 1 to entry: Data can be in different representations: material, analogue, digital, etc.

Note 2 to entry: Data is without interpretation and has no meaning; thus, some automated computation can be used to capture and process data.

3.1.2
information
structured, contextualized or processed data (3.1.1) that are endowed with meaning

Note 1 to entry: Information may be in different representations: material, analogue, digital, etc.

3.1.3
information security
preservation of confidentiality, integrity and availability of information (3.1.2)

Note 1 to entry: In addition, other properties, such as authenticity, accountability, non-repudiation, and reliability can also be involved.

[SOURCE: ISO/IEC 27000:2018 [1], 3.28]

3.1.4
knowledge
synthesis of multiple sources of information (3.1.2) over time to create conceptual frameworks, theories and axioms

Note 1 to entry: Knowledge gives context through experience, values and insight.

3.1.5
metadata
data (3.1.1) defining and describing other data

Note 1 to entry: Metadata is actually information (3.1.2) about other information.

[SOURCE: ISO 37156:2020 [2], 3.2.5, modified – Note 1 to entry has been added.]

3.2 Urban

3.2.1
citizen
person who lives or works in a city (3.2.2) or visits a city

Note 1 to entry: A citizen who lives in a city is a resident.

3.2.2
city
urban area or place with a name and defined geographical boundaries considered together with its citizens

Note 1 to entry: A city is sometimes referred to as a municipality or a local government.

Note 2 to entry: An urban area sometimes grows beyond the administrative area of the city administration.

Note 3 to entry: Cities may refer to any geographically located population.

Note 4 to entry: A city is, potentially, a self-organizing system.

3.2.3
city administration
organizations (3.3.16) for governing, managing and operating a city (3.2.2)

Note 1 to entry: Operating means being responsible for the running of a city (3.2.2).

3.2.4
city sustainability
sustainability (3.3.27) of a city (3.2.2)

Note 1 to entry: The sustainability of city might be based on five main aspects: economic, social, environmental, governance and cultural.

Note 2 to entry: There are six purposes of city sustainability in ISO 37101:2016: attractiveness, preservation and improvement of the environment, resilience, responsible resource use, social cohesion, and well-being.

3.2.5
community
group of people with commonality such as norms, religion, values, customs, identity, occupation or sense of place situated in a given geographical area

Note 1 to entry: A city (3.2.2) is a type of community.

Note 2 to entry: Communities can also function partially or completely online – as virtual communities.

3.3 Systems approach

3.3.1
architecture
complex of ideas and decisions about an entity (3.3.7) that are essential for achieving desired properties of the entity

Note 1 to entry: Architecture and desired properties of the entity define governing principles for the realization and evolution of this entity and its related life cycle processes.

Note 2 to entry: Desired properties and emerging properties are not always the same.

Note 3 to entry: Architecture is an inherent property of the entity. Architecture cannot be "extracted" from the entity; but one can give a description of the architecture.

Note 4 to entry: The entity that is concerned shall be specified, e.g. architecture of a city, architecture of a system.

3.3.2
artefact
object (3.3.15) made, directly or indirectly, by creative human work

3.3.3
beneficiary
stakeholder (3.3.26) who gains some value from the entity's (3.3.7) existence

Note 1 to entry: The entity that is concerned shall be specified, e.g. beneficiary of a city, beneficiary of a system.

[SOURCE: ISO/PAS 19450:2015 [3], 3.6, modified – "functional" replaced by "some", "system's operation" replaced by "entity's existence", Note 1 to entry has been added.]

3.3.4
capability
ability of an entity (3.3.7) to do something at the agreed level of performance

Note 1 to entry: The entity that is concerned shall be specified, e.g. capability of a city, capability of a system.

3.3.5
characteristic
abstraction of a property (3.3.18) of an object (3.3.15) or of a set of objects

EXAMPLE Weight.

Note 1 to entry: Characteristics are used for describing objects (3.3.15).

[SOURCE: ISO 1087-1:2000 [4], 3.2.4]

3.3.6
continuity
capability (3.3.5) of an entity (3.3.7) to plan for and respond to conditions, situations and events in order to continue operations at an acceptable predefined level following a disruption

Note 1 to entry: Continuity is the more general term for operational and business continuity to ensure an organization's ability to continue operating outside normal operating conditions. It applies not only to for-profit companies, but to organizations of all types, such as non-governmental, public interest and governmental.

Note 2 to entry: Continuity sometimes refers to the degree to which the IT service is provided under all foreseeable circumstances, including mitigating the risks resulting from interruption to an acceptable level.

Note 3 to entry: Continuity often refers to a state of stability and the absence of disruption.

Note 4 to entry: Continuity may refer to capability of an organization to continue the delivery of products or services at acceptable predefined levels following a disruption.

Note 5 to entry: Acceptable refers to satisfy stakeholders' expectations within prescribed specifications.

Note 6 to entry: The entity that is concerned shall be specified, e.g. continuity of a service, continuity of a system.

[SOURCE: ISO 22300:2018 [5], 3.24, modified – Note 6 to entry has been added.]

3.3.7
entity
object (3.3.15) with an identity

3.3.8
environment
context determining the setting and circumstances of all influences on a system (3.3.29)

Note 1 to entry: The environment of a system includes developmental, technological, business, operational, organizational, political, economic, legal, regulatory, ecological and social influences.

Note 2 to entry: The system that is concerned shall be specified, e.g. environment of a city.

[SOURCE: ISO/IEC/IEEE 24748-1:2018 [6], 3.20, modified – Note 2 to entry has been added.]

3.3.9
identifier
character or string of characters, used to identify or name an object (3.3.15)

Note 1 to entry: The object that is concerned shall be specified, e.g. identifier of an object.

3.3.10
indicator
measurable representation of conditions or statuses by means of quantitative and qualitative terms in order to assess the value of the activities characterized, and the associated method

[SOURCE: ISO 21248:2019 [7], 3.37; ISO 14005:2019 [8], 3.4.7, modified – "expression (which may be numeric, symbolic or verbal) used to characterize activities (events, objects, persons) both in" has been replaced by "measurable representation of conditions or statuses by means of”.]

3.3.11
infrastructure
vital services (3.3.25) of a system (3.3.29) which serve many other system elements (3.3.30)

Note 1 to entry: The system that is concerned shall be specified, e.g. infrastructure of a city.

3.3.12
interoperability
ability of diverse entities (3.3.7) to work together for a specified purpose

3.3.13
life cycle
evolution from conception through to destruction

Note 1 to entry: A typical life cycle comprises several relatively stable periods called phases. The phases are not necessarily sequential.

Note 2 to entry: In accordance with ISO/IEC/IEEE TR 24748-1 [6], the typical system (3.3.29) life cycle phases include conception, development, production, utilization (operation), support, maintenance, retirement and destruction.

Note 3 to entry: Some system elements can be versionable; thus their life cycle may include refactoring (i.e. the creation of another version with its own life cycle) in addition to the typical system life cycle phases.

Note 4 to entry: The human-made entity that is concerned shall be specified, e.g. life cycle of a system, life cycle of a product, life cycle of a service or life cycle of a project.

[SOURCE: ISO/IEC 15288:2015 [9], 4.1.23, modified – "through retirement" has been replaced by "through to destruction"; Notes 1, 2, 3 and 4 to entry have been added.]

3.3.14
maintenance
process of keeping something in good condition

Note 1 to entry: There are four types of maintenance: corrective, preventive, predictive, adaptive.

Note 2 to entry: Maintenance is a stage of systems life cycle.

3.3.15
object
anything perceivable or conceivable

Note 1 to entry: Objects may be material (e.g. an engine, a sheet of paper, a diamond), immaterial (e.g. a conversion ratio, a project plan) or imagined (e.g. a unicorn).

[SOURCE: ISO 1087-1:2000 [4], 3.1.1]

3.3.16
organization
legal or administrative entity (3.3.7) with structure, policies, employees, roles and responsibilities to achieve some stated purposes

Note 1 to entry: The concept of organization includes, but is not limited to, sole-trader, company, corporation, firm, enterprise, authority, partnership, charity or institution, or part or combination thereof, whether incorporated or not, public or private.

3.3.17
platform
coherent set of services (3.3.25) for the particular problem space or subject field

3.3.18
property
attribute or quality of an object (3.3.15) or of a set of objects

Note 1 to entry: The object or a set of objects that is concerned shall be specified, e.g. property of an object.

3.3.19
repeatability
ability of an entity (3.3.7) to be easily adaptable to and/or adoptable into different systems (3.3.29) or environments (3.3.8)

Note 1 to entry: Scalability of an entity is one of the many aspects of repeatability. Sometimes both can be mentioned together as requirements for an entity.

Note 2 to entry: The term "repeatable" is associated with the concept expressed by the French word "tirage" in the sense of mass distribution of a unique product.

Note 3 to entry: The entity that is concerned shall be specified, e.g. repeatability of a solution.

3.3.20
reference architecture
template for solution architectures (3.3.1) which realizes a predefined set of requirements

Note 1 to entry: A reference model is the next higher level of abstraction to the reference architecture.

Note 2 to entry: A reference architecture uses its problem space or subject field reference model and provides a common (architectural) vision, a modularization and the logic behind the architectural decisions taken.

Note 3 to entry: There may be several reference architectures for a single reference model.

Note 4 to entry: A reference architecture is universally valid within a particular problem space or subject field.

Note 5 to entry: An important driving factor for the creation of a reference architecture is to improve the effectiveness of creating products, product lines and product portfolios by managing synergy, providing guidance, e.g. architecture principles and good practices providing an architecture baseline and an architecture blueprint, and capturing and sharing (architectural) patterns.

Note 6 to entry: In the field of software architecture or enterprise architecture, reference architecture provides a proven template for solution architectures in particular problem space, as well as a common vocabulary with which to discuss implementations, often with the aim of stressing commonality.

3.3.21
reference model
abstract framework for understanding concepts and relationships between them in a particular problem space or subject field

Note 1 to entry: A reference model is independent of the technologies, protocols and products, and other concrete implementation details.

Note 2 to entry: A reference model uses a concept system for a particular problem space or subject field.

Note 3 to entry: A reference model is often used for the comparison of different approaches in a particular problem space or subject field.

Note 4 to entry: A reference model is usually a commonly agreed document, such as an International Standard or industry standard.

Note 5 to entry: See also IEC 61970-2 [10], OASIS Reference Model for Service Oriented Architecture 1.0 [11], and the Togaf9 standard [12] .

[SOURCE: SAMARIN, A., 2009, [13], 14.2.16, 14.2.14, modified – "domain" has been replaced by "subject field".]

3.3.22
resilience
ability of an entity (3.3.7) to cope with undesired change

Note 1 to entry: The entity that is concerned shall be specified, e.g., resilience of a service.

3.3.23
safety
freedom from risk which is not tolerable

[SOURCE: ISO/IEC Guide 51:2014 [14], 3.14]

3.3.24
security
condition that results from the design and maintenance (3.3.14) of protective measures that ensure a state of inviolability from hostile acts or influences

[SOURCE: IEC Guide 120:2018 [15], 3.13, modified – "establishment" has been replaced by "design"; Note 1 to entry has been deleted]

3.3.25
service
group of one or more capabilities (3.3.4) of an entity (3.3.7) accessible using a prescribed interface

Note 1 to entry: The entity capabilities provide some value for its consumer.

Note 2 to entry: The service consumer and the service provider may belong to different ownership realms.

Note 3 to entry: The entity that is concerned shall be specified, e.g. service of a city.

3.3.26
stakeholder
person, group of persons or organization (3.1.4) having an interest in an entity

Note 1 to entry: The entity that is concerned shall be specified, e.g. stakeholder of a city.

[SOURCE: ISO/IEC/IEEE 42010:2011, 3.10, modified – "system" has been replaced by "entity", "individual" has been replaced by "person" and "or classes thereof" deleted, note 1 to entry has been added.]

3.3.27
sustainability
ability of a self-organizing system (3.3.29) to meet its needs as long as necessary

Note 1 to entry: Sustainability is the goal of sustainable development.

Note 2 to entry: The self-organizing system that is concerned shall be specified, e.g. sustainability of a city.

3.3.28
sustainable development
development that preserves or builds sustainability (3.3.27)

3.3.29
system
entity represented as a complex of interacting entities (3.3.7) organized as a whole which exhibits (as the result of interaction between the entities) some emergent characteristics indispensable to achieve one or more stated purposes

Note 1 to entry: Systems are different from structural assemblies.

Note 2 to entry: A system is sometimes considered as a product or as the services it provides.

Note 3 to entry: In practice, the term "system" is frequently qualified through the use of an associative noun to indicate the subject field, e.g. aircraft system. Sometimes the term system is substituted by a context-dependent synonym, e.g. aircraft, but such practice is deprecated since it potentially obscures the meaning from a system principles perspective.

Note 4 to entry: A system includes all of the associated equipment, facilities, materials, computer programs, firmware, technical documentation, services and personnel required for operations and support to the degree necessary for self-sufficient use in its intended environment. In some cases, the behaviour of a system cannot be explained in terms of the behaviour of its discrete parts.

Note 5 to entry: Typical characteristics of a system are:

a) systems are an idealization;

b) systems have multiple elements;

c) the components are interdependent;

d) systems are organized;

e) systems have emergent properties;

f) systems have a boundary;

g) systems are enduring;

h) systems affect and are affected by their environment;

i) systems exhibit feedback; and

j) systems have non-trivial behaviour.

3.3.30
system element
discrete entity (3.3.7) of a system (3.3.29) that contributes to one or more of its capabilities (3.3.4)

EXAMPLE: Hardware, software, data, persons, assistive animals, assistive robots, processes (e.g. processes for providing a service to users), procedures (e.g. operator instructions), facilities, materials, and naturally occurring entities or any combination thereof.

Note 1 to entry: In [16] "discrete" is defined as "individually separate and distinct". Something that is "individually separate and distinct" can change depending on the context, e.g. depending on the level of detail that one is looking at, on the perspective, etc.

Note 2 to entry: A system element can contain other system elements.

3.3.31
systems thinking
complex of synergistic analysis and synthesis skills used to improve the capability of identifying and understanding systems, predicting their behaviours, and devising modifications to them in order to produce desired effects

Note 1 to entry: These skills work together as a system.

Note 2 to entry: These skills are the following:

1) recognizing interconnections;
2) identifying and understanding feedback;
3) understanding system structure;
4) differentiating types of stocks, flows, variables;
5) identifying and understanding non-linear relationships;
6) understanding dynamic behaviour;
7) reducing complexity by modelling systems conceptually;
8) understanding systems at different scales.

[SOURCE: Ross D. Arnold, 2015 [17] , modified – "analytic" is replaced by "analysis and synthesis".]

3.3.32
system approach idea
idea that considers that functional and structural engineering, system-wide interfaces and system (3.3.29) properties become more and more important due to the increasing complexity, convergence and interrelationship of technologies

[SOURCE: IEC AC/7/2004 [18], modified – The word "compositional" has been removed.]

3.3.33
systems approach
particular way of applying systems thinking (3.3.31)

Note 1 to entry: There are many systems approaches; IEC develops its own systems approach for systems works.

3.3.34
system of systems
SoS (deprecated)
representation of a system (3.3.29) as a complex of managerially and operationally independent systems which are coordinated together for a period of time to achieve one or more commonly stated purposes

Note 1 to entry: Each constituent system is a useful system by itself, having its own management, goals, and resources, but coordinating together within the system of systems to provide one or more emergent capabilities of the system of systems.

3.4 Digital transformation

3.4.1
digitally coordinated system
digital system
system (3.3.29) in which the digital presentation (3.4.2) of system elements (3.3.30), system and its element properties, architecture views on system and its elements, and relationships between everything mentioned above is a dominant consideration

Note 1 to entry: The qualifier "digitally coordinated" points to the essential characteristic of such systems. The same logic is used for the definition of socio-technical systems (importance of interactions between people and technology in workspaces) and the definition of cyber-physical systems (importance of interactions between physical and computing components).

Note 2 to entry: For some objects their digital representation is secondary to their physical representation, for some objects their digital representation is primary to their other representations and for some objects only their digital representation exists.

Note 3 to entry: A digitally coordinated system uses digital representations throughout its life cycle to be architected, built, governed, managed, operated, evolved, decommissioned and destroyed.

Note 4 to entry: Digital representations are enablers for applying innovative digital and other technologies and facilitating deployment of technologies. For example, an impact of a particular new technology can be objectively evaluated with the use of digital representations.

Note 5 to entry: The digitally coordinated system captures, analyses and helps to align different views from different stakeholders, and thus enables the progress. Examples of such views are physical, social, ethical, financial, economic, legal, cultural, informational, etc.

3.4.2
digital representation

formal, explicit, computer-readable and computer-executable description that allows a computer to perform various operations on an object (3.3.15)

EXAMPLE Digital representation of an object may be a computational model of this object.

Note 1 to entry: Digital representation of an object may be primary or secondary relative to other representations of this object.

Note 2 to entry: The object that is concerned shall be specified, e.g. digital representation of a building.

3.4.3
digital transformation

DT

DX, Japan

systemic increase in the level of orderliness of a complex entity (3.3.7) as a digitally coordinated system (3.3.29)

Note 1 to entry: Orderliness is about increased coherence and reduced complexity.

Note 2 to entry: DT improves various intrinsic characteristics of this entity and their match with various views on the systems. Examples of such intrinsic characteristics are trustworthiness, low cost to maintain, reduced time-to-market, etc.

Note 3 to entry: DT is not a project, a programme or a product, it never ends, it is a life style [19].

Note 4 to entry: The complex entity that is concerned shall be specified, e.g. DT of a city.

3.4.4
digital twin

totality of digital representations (3.4.2), algorithms and recorded facts pertinent to an entity (3.3.7) over its life cycle (3.3.13)

Note 1 to entry: The digital twin of an entity is used by a system, in which this entity is a system element, to manage and operate this entity.

Note 2 to entry: The entity that is concerned shall be specified, e.g. digital twin of a street.

3.4.5
smart city

city (3.2.2) which is permanently transforming itself as a digitally coordinated system to align the various, primarily citizens', views of the city

Note 1 to entry: Citizens' views may be different in each city: better sustainability, higher level of resilience, accelerated economic development, etc.

Note 2 to entry: The complexity of cities requires treating them as digitally coordinated systems.

Note 3 to entry: Building smart city is DT of a city.

Note 4 to entry: Each smart city carries out its DT at its own pace.

Note 5 to entry: The shortest form of the definition is "city built as a digital system".

3.5 Architecting

3.5.1
architecture view
work product expressing the architecture (3.3.1) of a system (3.3.29) from the perspective of specific system concerns

[SOURCE: ISO/IEC/IEEE 42010:2011, 3.5]

3.5.2
model kind
conventions for a type of modelling

EXAMPLE Examples of model kinds include data flow diagrams, class diagrams, Petri nets, balance sheets, organization charts and state transition models.

[SOURCE: ISO/IEC/IEEE 42010:2011, 3.9]

3.5.3
architecture viewpoint
work product establishing the conventions for the construction, interpretation and use of architecture views (3.5.1) to frame specific system concerns

3.5.4
concern
<system> interest in a system relevant to one or more of its stakeholders

Note 1 to entry: A concern pertains to any influence on a system in its environment, including developmental, technological, business, operational, organizational, political, economic, legal, regulatory, ecological and social influences.

[SOURCE: ISO/IEC/IEEE 42010:2011, 3.7]

3.5.5
Smart Cities Reference Architecture
SCRA
architecture (3.3.1) of an idealized (reference) smart city (3.4.5) which realizes a predefined set of requirements and is used as a template for architectures (3.3.1) of other smart cities

Note 1 to entry: The SCRA can be considered as a framework (coherent set of ideas, principles, agreements or rules which provides the basis or outline for something intended to be more fully developed at a later phase) for other smart cities.

3.5.6
Smart Cities Reference Architecture Methodology
SCRAM
coherent set of methods for developing the SCRA (3.5.5)

Note 1 to entry: The SCRAM is necessary for effective tailoring of the SCRA for other smart cities.

Note 2 to entry: The SCRAM can be considered as an architecture for the SCRA.

3.5.7
SCRA view
architecture view (3.5.1) for a city (3.2.2) as a system (3.3.29)

3.5.8
SCRA model
work product expressing partially the architecture (3.3.1) of a city (3.2.2) as a system (3.3.29) for the needs of some SCRA views (3.5.7)

Note 1 to entry: A SCRA model can be a system element of the city as a system.

3.5.9
SCRA artefact

work product expressing an entity made by creative human work to be used in some SCRA models (3.5.8)

Note 1 to entry: A SCRA artefact can be a system element of the city as a system.

3.5.10
SCRAM viewpoint

work product establishing the common conventions for the construction, interpretation and use of SCRA views (3.5.7)

EXAMPLE Problem space viewpoint, integration viewpoint, engineering viewpoint, technology viewpoint.

Note 1 to entry: A SCRAM viewpoint is an adaptation of architecture viewpoint (3.5.3).

3.5.11
SCRAM model-type

work product establishing the common conventions for the construction, interpretation and use of SCRA models (3.5.8)

Note 1 to entry: A SCRAM model-type is an adaptation of model kind (3.5.2).

3.5.12
SCRAM artefact-type

work product establishing the conventions for the construction, interpretation and use of SCRA artefacts (3.5.9)

4 Abbreviated terms

AIIM	Association for Intelligent Information Management
AI	Artificial Intelligence
API	Application Programming Interface
APQC	American Productivity & Quality Centre
BPM	Business Process Management
BPMN	Business Process Modelling and Notation
DAMA	DAta Management Association
DLP	Data Loss Protection
DMN	Decision Model and Notation
DT	Digital Transformation
EDA	Event Driven Architecture
EPN	Event Processing Network
GDPR	General Data Protection Regulations
ICRC	International Committee of the Red Cross
IDEF0	Integration DEFinition language 0
IDL	Interface Definition Language
IFD	Information Flow Diagram
IFRC	International Federation of Red Cross and Red Crescent Societies
IREB	International Requirements Engineering Board
IT	Information Technology
ITIL	Information Technology Infrastructure Library

ITSM	Information Technology Service Management
LLUC	Low Level Use Case
MBSE	Model-Based Software Engineering
M2M	Machine-2-Machine
MVP	Minimum Viable Product
NIST	National Institute of Standards and Technology
OECD	Organisation for Economic Co-operation and Development
PaaS	Platform as a Service
PCF	Process Classification Framework
PEST	Political, Economic, Social and Technological)
SIEM	Security Information and Event Management
SDO	Standard Development Organisation
SRD	Systems Reference Deliverable
SRG	Systems Resource Group
SWOT	Strengths, Weaknesses, Opportunities, and Threats
UI	User Interface
UML	Unified Modelling Language
UN	United Nations
URL	Universal Resource Locator
WHO	World Health Organization

5 Linking the IEC SyC Smart Cities scope with the SCRAM and SCRA

5.1 Methodology

Knowing that the SCRAM is actually a meta reference architecture, i.e. an architecture of reference architecture, the SCRAM shall be in conformance with ISO/IEC/IEEE 42010:2011. Thus the description of the SCRAM shall be expressed via architecture viewpoints, architecture views, model kinds and models.

For linking the IEC SyC Smart Cities scope with the SCRAM and SCRA, an architecture viewpoint "justification" is defined and a view "justification" is created. These viewpoint and view are provided for all the readers of the SCRAM to explicitly and step-by-step explain how the scope of IEC SyC Smart Cities and other pertinent information outline the SCRAM. This viewpoint comprises a few simple model kinds which are: text and keywords, nomenclatures or dependency matrixes. These model kinds are listed below.

- Problem space description (5.2), which is the analysed scope of IEC SyC Smart Cities.
- Problem space specifics (5.3) nomenclature, which is derived from the general knowledge of smart cities system domain.
- Stakeholders classification (5.4) which is based on the summary of the SRG work on systems approach.
- Stakeholders (5.5) nomenclature (individuals, teams, organizations having an interest in smart cities), which is derived from the problem space description.
- Dependency matrix (5.6) "Stakeholders classification" versus "Stakeholders".
- Dependency matrix (5.7) "Problem space description" versus "Stakeholders".
- Stakeholders' aggregated concerns (5.8) nomenclature, which is derived from the stakeholders. These concerns are aggregated and serve as initial requirements.

- Mission statement (5.9), which outlines what problem IEC SyC Smart Cities is going to solve (see scope and problem space specifics) for whom (see stakeholders).
- Dependency matrix (5.10) "Problem space description" and "Problem space specifics" versus "Mission statement".
- Dependency matrix (5.11) "Stakeholders" versus "Mission statement" and "Stakeholders' aggregated concerns".
- Vision statement (5.12) which outlines a solution space (variations of potential solutions) in accordance with the mission statement and satisfies the stakeholders' aggregated concerns.
- Dependency matrix (5.13) "Mission statement" and "Stakeholders' aggregated concerns" versus "Vision statement".
- Strategic goals (5.14) nomenclature, which are measurable results mandatory to achieve the vision statement.
- Dependency matrix (5.15)"Vision statement" versus "Strategic goals".
- Solution space constraints (5.16) nomenclature, which limit the variety of potential solutions by imposing some choices.
- Essential desirable characteristics (5.17) nomenclature of the solution space (thus all potential solutions), which address the strategic goals and the solution space constraints.
- Dependency matrix (5.18) "Strategic goals" and "Solution space constraints" versus "Essential desirable characteristics".
- Architecture principles (5.19) nomenclature guarantees that potential solutions will possess the essential desirable characteristics.
- Dependency matrix (5.20) "Essential desirable characteristics" versus "Architecture principles".

The dependency matrixes are relationships in Figure 1.

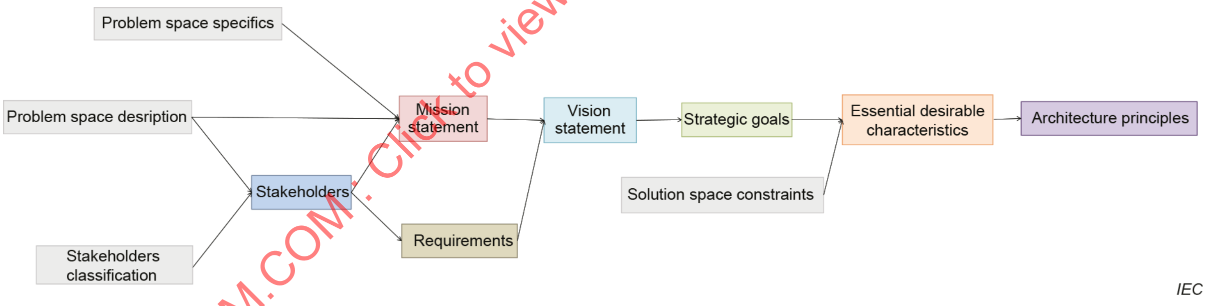


Figure 1 – Relationships within the "justification" viewpoint

5.2 Problem space description

IEC SyC Smart Cities has the following scope [20][1], approved by the IEC Standardization Management Board. This scope is considered as the problem space description.

[1] Numbers in square brackets refer to the Bibliography.

To foster the development of standards in the field of electrotechnology to help with the integration, interoperability and effectiveness of city systems.

NOTE 1 This will be done:

- by promoting the collaboration and systems thinking between IEC TCs, the SyCs and other SDOs in relation to city system standards;
- by undertaking systems analysis to understand the needs for standards and assess new work item proposals (NWIPs) related to city systems;
- by developing systems standards where needed and by providing recommendations to existing SyCs, TCs/SCs and other SDOs.

NOTE 2 Overall common city goals include, for example, sustainable development, efficiency, resilience, safety and support for citizens' engagement and participation. However, an individual city will follow its own approach.

NOTE 3 "Cities" refers to any geographically located population.

5.3 Problem space specifics

NOTE In 5.3, the text fragments in bold will be used in some dependency matrix.

All smart city programmes and projects pursue many common goals including sustainable development, better efficiency, resilience, safety and wider support for citizen's engagement and participation. However, the work of the Standardization Evaluation Group on Smart Cities (SEG 1) – which is the predecessor of SyC Smart Cities and now disbanded – has shown that **each individual city tends to follow its own approach in smart cities programmes and projects.**

Often city governance bodies are confused by the numerous technology activists who are very vocal on various smart cities forums even though **city systems cannot be reduced to just big data and IoT** (see [21]).

Another result of the SEG 1 is that there is no common understanding even in standardization bodies about smart cities. About 20 reference models collected by the SEG 1 are incompatible, so there is a **huge demand for a common understanding**.

Experience of SyC Smart Cities members in various smart cities projects has shown that the **current implementation practices of smart cities are rather disjointed**, namely:

- smart cities programmes and projects are, primarily, local initiatives;
- smart cities programmes and projects are considered as technology projects;
- numerous smart cities interest groups are, primarily, clubs;
- efforts for the development of a common vision are insufficient; and
- typical financing patterns do not promote a common vision, namely the government is funding (to some extent) some cities, which engage technological companies, and the government is funding some technological companies, which engage cities.

As a result, there is **no agreed basis for efficient and effective cooperation and coordination** between different smart cities programmes and projects. There is a lot of work duplication, developed solutions are not reusable, and the same mistakes are widely repeated.

A look at the number of cities within different population categories indicates the imperative of bringing standardization into this **complex paradigm of cities,** which are systems of systems leveraging networks of networks to bring operational efficiency, sustainability and resilience to the city infrastructure to improve its citizens' quality of life.

City systems are multi-disciplinary and use multiple technologies; therefore, **city systems shall be carefully architected** to allow all technologies to work together.

5.4 Stakeholders classification

NOTE In 5.4, the text fragments in bold will be used in some dependency matrix.

The summary of the SRG work on systems approach recommends the development of a classification of stakeholders to facilitate the process of finding the actual stakeholders. The stakeholders classification is as follows.

- **Primary beneficiary**: Stakeholders that benefit directly from the particular system.
- **Secondary beneficiary**: Stakeholders that benefit indirectly from the particular system.
- **Tertiary beneficiary**: Stakeholders that benefit somehow from the particular system.
- **Owner**: Stakeholders that own the particular system.
- **Designer**: Stakeholders that are part of the ecosystem of those participating in the architecting of the particular system.
- **Builder**: Stakeholders that are part of the ecosystem of those participating in the construction of the particular system.
- **Maintainer**: Stakeholders that are part of the ecosystem of those participating in the maintenance of the particular system.
- **User**: Stakeholders that are part of the ecosystem of those participating in the use of the particular system for their needs.
- **Operator**: Stakeholders that are part of the ecosystem of those participating in the operations of the particular system.
- **Tester**: Stakeholders that are part of the ecosystem of those participating in the testing of the particular system.

5.5 Stakeholders and their estimated concerns

NOTE In 5.5, the text fragments in bold will be used in some dependency matrix.

Citizens are the indirect beneficiaries. They will profit from this set of city system standards indirectly. It will be for each city to collect the citizens' concerns and treat them systematically in accordance with this set of city system standards, along with the related tools and guiding materials. Below are examples of citizens' concerns (the complete set will be referred to as "full city system functionality" in the remainder of this document):

- adequate water supply;
- assured electricity supply;
- sanitation, including solid waste management;
- efficient urban mobility and public transport;
- affordable housing, especially for the poor;
- widespread and transformative use of data and technology;
- good governance and citizen participation;
- sustainable environment;
- safety and security of citizens, particularly women, children and the elderly;
- affordable healthcare for everyone;
- modern education for children and adults;
- attractive for business.

Business (local) is the indirect beneficiaries. Their primary expected concern is the following:

- attractive for business.

City governance bodies are responsible for the prioritization and execution of city systems improvement activities. City governance bodies' expected concerns are the following:

- methods to understand needs of city systems stakeholders;
- critical success factor based methods to estimate the complexity, cost and benefit for implementation of smart cities programmes and projects to determine the returns on those investments directly and indirectly from a forecast to actual series of measures;
- good business practices and world-class knowledge regarding smart cities to enable them to properly manage and execute smart cities programmes and projects;
- enabling collaboration and coordination with other smart cities programmes and projects to improve the quality, share the cost, reduce time-to-value and manage associated risks.

Smart city architecture teams are responsible for understanding smart cities and describe them in a common structure by change to describe this set of city system standards to the unique needs of the particular city. Smart city architecture teams' expected concerns are the following:

- quickly understandable set of city system standards;
- quickly mastered set of city system standards;
- easy to use supporting tools;
- wide use of this set of city system standards.

Standards development groups (IEC TCs, SyCs and other SDOs) are responsible for formal definition of some functional elements and their interfaces which are outlined by this set of city system standards. Also, this "set of instruments" provides enough information on how to define interfaces as APIs and how to describe functional elements. Standards development groups expected concerns are the following:

- easy to comprehend and to apply set of city system standards;
- easy to use interoperable supporting tools;
- wide use of this set of city system standards.

Vendors are responsible for implementing some functional elements and their interfaces. Vendors' expected concerns are the following:

- quickly understandable set of city system standards to identify opportunities for developing products which can be used by cities following the SCRAM and SCRA.

Investors are responsible for taking informed decisions about their investments and liabilities in city systems infrastructure. At the moment cities are missing out on significant investments that should be available to them because of: a) the potential to mitigate huge potential economic losses from disasters, b) growing popularity of sustainable energy investments prioritizing investment ahead of other areas such as smart cities infrastructure, and c) the high risks of existing city bonds.

This set of city system standards can be used to benchmark various smart cities related work. Also, applying this set of instruments brings good business practices and world-class knowledge to smart cities programmes and projects. Investors' expected concerns are the following:

- wide use of this set of city system standards, which will act as an investment multiplier.

SyC Smart Cities' concerns are the following:

a) addressing by-design city system quality characteristics such as
   - interoperability,
   - safety,

- security (including confidentiality, integrity and availability),
- privacy,
- resilience,
- simplicity,
- low cost of operation,
- short time-to-value,
- combining diversity and uniformity,
- self-referential, and
- system-life-cycle;

b) covering the whole life cycle of city systems.

5.6 Dependency matrix "Stakeholders classification" versus "Stakeholders"

Figure 2 provides this dependency matrix.

ROWS are sources (Stakeholders Classification) vs. COLUMNS are destinations (Stakeholders)	Citizens	City governance bodies	Smart City architecture teams	Standards development groups	Vendors	Investors	SyC Smart Cities	Business
Primary beneficiary				X				
Secondary beneficiary			X		X			
Tertiary beneficiary	X	X						x
Owner							X	
Designer					X		X	
Builder						X	X	
Maintainer				X			X	X
User			X	X	X			
Operator								X
Tester								X

IEC

Figure 2 – A dependency matrix "Stakeholders classification" versus "Stakeholders"

5.7 Dependency matrix "Problem space description" versus "Stakeholders"

Figure 3 provides this dependency matrix.

ROWS are sources (Problem space description) vs. COLUMNS are destinations (Stakeholders)	Citizens	City governance bodies	Smart City architecture teams	Standards development groups	Vendors	Investors	SyC Smart Cities	Business
foster the development of standards				X			X	
integration, interoperability and effectiveness				X			X	
collaboration and systems thinking			X				X	
IEC TCs, SyCs and other SDOs				X			X	
city system standards					X			
understand the needs						X		
city systems			X					
common city goals		X						
sustainable development, efficiency, resilience, safety			X					
support for citizens' engagement and participation	X							X
individual city will follow its own approach		X						X
any geographically located population	X							X

IEC

Figure 3 – A dependency matrix "Problem space description" versus "Stakeholders"

5.8 Stakeholders' aggregated concerns

For the purpose of this document, only stakeholders' aggregated concerns are pertinent (see Figure 5). They are derived from 5.7 and listed below:

- full city system functionality;
- quickly understandable set of city system standards;
- quickly mastered set of city system standards;
- easy to use supporting tools;
- this set of instruments shall help stakeholders to estimate, plan, manage and execute smart cities programmes and projects;
- ability to share experience, collaborate and coordinate among smart cities programmes and projects which use this set of city system standards;
- wide use of this set of city system standards;
- addressing by-design city system quality characteristics;
- covering the whole life cycle of city systems.

5.9 Mission statement

At present IEC SyC Smart Cities does not have a formally agreed mission statement. However, the following mission statement could be derived from the agreed scope.

> Systematically develop, together with IEC TCs, SyCs, other SDOs, a coherent and highly useful set of city system standards for efficient implementation any city system.

Here is clarification of what is meant by the key phrases of this mission statement.

- "systematically develop" – based on the summary of the SRG work on systems approach.

- "set of city system standards" – a comprehensive collection of standards shall cover all aspects of cities.
- "together with" – organizations which need to contribute to this set of city system standards are able to easily master how to contribute.
- "coherent" – all standards for city systems shall be compatible and developed under the same guidance.
- "highly useful" – in order to ensure that this set of city system standards are really useful, it is vital to identify all organizations which are working and using city systems and ensure that the standards developed address needs and requirements of those organizations.
- "efficient implementation" – strive for improving quality, saving money and reducing time-to-market for implementations.
- "any city system" – this set of city system standards shall be flexible and extensible to be useful for all types of city.

5.10 Dependency matrix "Problem space description" and "Problem space specifics" versus "Mission statement"

Figure 4 provides this dependency matrix.

ROWS are sources (Problem space description, Problem space specific) vs. COLUMNS are destinations (Mission statement)	systemically develop	set of city system standards	together with	coherent	highly useful	any city system
foster the development of standards			X			
integration, interoperability and effectiveness	X				X	
collaboration and systems thinking	X		X			
IEC TCs, SyCs and other SDOs			X			X
city system standards		X		X		
understand the needs					X	
city systems					X	X
common city goals					X	
sustainable development, efficiency, resilience, safety	X				X	
support for citizens' engagement and participation			X			
individual city will follow its own approach						X
any geographically located population						X
each individual city tends to follow its own approach						X
though city systems cannot be reduced to just big data and IoT	X			X		
huge demand for a common understanding		X				X
current implementation practices of Smart Cities are rather disjointed		X		X		
no agreed basis for efficient and effective cooperation and coordination					X	X
complex paradigm of cities	X			X		
city systems must be carefully architected	X					X

IEC

Figure 4 – A dependency matrix "Problem space description" and "Problem space specifics" versus "Mission statement"

5.11 Dependency matrix "Stakeholders" versus "Mission statement" and "Stakeholders' aggregated concerns"

Figure 5 provides this dependency matrix.

ROWS are sources (Stakeholders) vs. COLUMNS are destinations (Mission statement, Stakeholders' aggregated concerns)	systemically develop	set of city system standards	together with	coherent	highly useful	any city system	Full city system functionality	Quickly understandable set-of-city-standards	Quickly mastered set-of-city-standards	Easy to use supporting tools	This set-of-city-standards must help stakeholders to estimate, plan, manage and execute Smart Cities programmes and projects	Ability to share experience, collaborate and coordinate among Smart Cities programmes and projects which use this set-of-instruments	Wide usage of this set-of-instruments	Addressing by design city system quality characteristics	Covering the whole life cycle of city systems
Citizens						X	X								
City governance bodies					X	X	X				X	X	X		
Smart City architecture teams			X	X	X			X	X	X			X		
Standards development groups	X	X						X		X			X		
Vendors				X				X		X			X		
Investors				X				X					X		
SyC Smart Cities	X		X		X	X								X	X
Business						X	X					X			

IEC

Figure 5 – A dependency matrix "Stakeholders" versus "Mission statement" and "Stakeholders' aggregated concerns"

5.12 Vision statement

At present IEC SyC Smart Cities does not have a formally agreed vision statement. However, the following vision statement could be derived from the agreed scope:

> Transparently developed set of city system instruments (around the detailed smart cities Reference Architecture) which are easily adaptable for the unique needs of any city system.

Here is clarification of what is meant by the key phrases of this vision statement.

- "transparently developed" – thus everyone can understand the logic of IEC SyC Smart Cities work, validate this work and add to or further develop some parts of its work, if necessary.
- "set of city system instruments" – a coherent collection of methodological, domain-specific, informational, software-intensive, standard and informative tools.
- "Smart Cities Reference Architecture" – commonly-agreed architecture which all smart cities can use as a template for solution architectures of their city systems because the key decisions taken in the Smart Cities Reference Architecture (SCRA) will guarantee that future city systems (built in accordance with this SCRA) will possess by-design the essential quality characteristics, e.g. security and privacy.

- "detailed" – it is necessary to provide enough scope and enough decomposition of the SCRA to reach a level of detail which is suitable for effective standardization of city systems – functional blocks may be implementable as black-boxes and interfaces between them shall be not too complex.
- "easily adaptable" – although there are a lot of communalities between cities, each city always has its own unique features. Thus, city systems will be able to use many of common system elements and easily intermix them with their own system elements.
- "unique needs of any city system" – the set of instruments shall be flexible and extensible to be useful for all types of city.

5.13 Dependency matrix "Mission statement" and "Stakeholders' aggregated concerns" versus "Vision statement"

Figure 6 provides this dependency matrix.

ROWS are sources (Mission statement, Stakeholders' aggregated concerns) vs. COLUMNS are destinations (Vision statement)	Transparently developed	Set of city system instruments	Smart Cities Reference Architecture	Detailed	Easily adaptable	Unique needs of any city system
systemically develop	X					X
set of city system standards		X				
together with	X		X	X		
coherent		X	X			
highly useful				X		X
any city system					X	X
Full city system functionality		X		X		
Quickly understandable set-of-city-standards	X			X		
Quickly mastered set-of-city-standards	X			X		
Easy to use supporting tools		X			X	
This set-of-city-standards must help stakeholders to estimate, plan, manage and execute Smart Cities programmes and projects		X				X
Ability to share experience, collaborate and coordinate among Smart Cities programmes and projects which use this set-of-instruments	X	X				
Wide usage of this set-of-instruments		X				X
Addressing by design city system quality characteristics			X			X
Covering the whole life cycle of city systems			X			X

IEC

Figure 6 – A dependency matrix "Mission statement" and "Stakeholders' aggregated concerns" versus "Vision statement"

5.14 Strategic goals

NOTE 1 In 5.14, the text fragments in bold will be used in some dependency matrix.

At present, IEC SyC Smart Cities does not have formally agreed strategic goals. However, the following strategic goals could be derived from the agreed scope.

- Commonly-agreed **multidisciplinary concept system** to enable all the stakeholders to use the same "language".

  NOTE 2 Such a concept system is developed in separate documents of IEC SyC Smart Cities.

- Commonly-agreed Smart Cities Reference Architecture Methodology (**SCRAM**) to make transparent and understandable how the Smart Cities Reference Architecture (SCRA) has been developed.
- Commonly-agreed **SCRA** to serve as an easily tailorable template for architecting and implementing any city system.
- The continually updated and extended Smart Cities Implementation Manual (**SCIM**)[2] on how to tailor the SCRAM and SCRA to easily address unique needs of any city system.
- **Map of existing standards** to the SCRA to quickly navigate to find standards pertinent for city systems.

  NOTE 3 Such mapping is developed in separate documents of IEC SyC Smart Cities.
- Map of **potential standards** to the SCRA to list standards to be developed for city systems.

  NOTE 4 Such mapping is developed in separate documents of IEC SyC Smart Cities.
- Necessary **tools to simplify adoption** of the SCRAM, SCRA, SCIM and other deliverables.
- Practically **all smart cities programmes and projects use SCRAM, SCRA, SCIM** and related tools.

5.15 Dependency matrix "Vision statement" vs "Strategic goals"

Figure 7 provides this dependency matrix.

ROWS are sources (Vision statement) vs. COLUMNS are destinations (Strategic goals)	Multidisciplinary concept system	SCRAM	SCRA	SCIM	Map of city standards	Tools which simplify adoption	All smart cities use the SCRAM, SCRA and SCIM
Transparently developed		X	X				
Set of city system instruments	X	X	X	X	X		
Smart Cities Reference Architecture		X	X				
Detailed		X	X				
Easily adaptable	X	X	X	X			
Unique needs of any city system						X	X

IEC

Figure 7 – A dependency matrix "Vision statement" vs "Strategic goals"

5.16 Solution space constraints

NOTE In 5.16, the text fragments in bold will be used in some dependency matrix.

The IEC SyC Smart Cities solution space constraints are the following.

- Adopting and adapting the summary of the **SRG work on systems approach** (see Annex B).
- Although the IEC SyC Smart Cities responsibility is mainly regarding electrotechnical aspects of smart cities, the SCRA should cover **city systems as a whole**. All the system elements identified by the SCRA will be sorted among various SDOs, including the IEC, depending on the nature of those system elements.

[2] SCIM is conceptualized; however, its best form is still to be chosen. One of the options is an online IEC Guide.

- Working together with the pertinent stakeholders, including IEC/TCs, SyCs, other SDOs, cities, and various bodies of knowledge implies that **alignment with existing knowledge** shall be enforced.
- **Maximize the use of digital technologies**.

5.17 Essential desirable characteristics of the solution space

NOTE In 5.17, the text fragments in bold will be used in some dependency matrix.

The nomenclature of the essential desirable characteristics is the following.

- **Be flexible**: Because it is not possible to anticipate all unique needs of all smart cities, the SCRAM, SCRA and SCIM shall be able to allow their unlimited extension and take into account both the diversity of each city and the commonalities between all cities.
- **Be explicit**: The set of instruments shall explain to any stakeholder how future implementations (which are based on the reference architecture) can address his/her concerns and change his/her personal, professional and social life for the better.
- **Be systematic**: The set of instruments provides a common approach for architecting and engineering city systems, so different people in similar situations can find similar solutions or propose innovations that will be valuable for others as well.
- **Be efficient**: The set of instruments helps stakeholders, programmes and projects to collaborate (e.g. co-design and co-implementation) and coordinate their efforts via common agreements (i.e. standards) on various system elements (e.g. services, interfaces, data), common vision, common tools, common products, etc. This leads to easy repetition of good solutions among all smart cities, easy sharing of business practices in building, running and evolving smart cities and removing entry barriers for local IT expertise and start-ups.
- **Be technology neutral**: Because of technology progress, many various (and unknown right now) intelligent devices (or "Things" from the Internet of Things) and digital technologies will progressively automate, improve and drastically change various aspects of smart cities functioning including planning, execution, monitoring, prediction, and optimization of flows.
- **Be trustworthy**: High level of trustworthiness (including security, privacy, safety, reliability, and resilience) is mandatory by-design and by-default.
- **Be digital**: Smart city is a city which is (re)built as a digital repeatable system.
- **Be effective**: The SCRAM, SCRA and SCIM shall provide the level of granularity which is low enough for efficient implementation of city systems and high enough for standardization purposes.
- **Be social:** smart cities shall bring social and economic innovations to citizens, business and city administration. (see [22]).

5.18 Dependency matrix "Strategic goals" and "Solution space constraints" versus "Essential desirable characteristics"

Figure 8 provides this dependency matrix.

ROWS are sources (Strategic goals, Solution space constraints) vs. COLUMNS are destinations (Essential desirable characteristics)	Be flexible	Be explicit	Be systematic	Be efficient	Be technology neutral	Be trustworthy	Be digital	Be effective	Be social
Multidisciplinary concept system			X		X				X
SCRAM				X	X	X	X		
SCRA				X			X		
SCIM	X				X			X	
Map of city standards		X	X						
Tools which simplify adoption		X							
All smart cities use the SCRAM, SCRA and SCIM	X			X					
IEC Systems Approach		X							
The whole city system	X	X		X		X		X	
Alignment of existing knowledge	X			X					X
Maximize the usage of digital technologies			X				X		

IEC

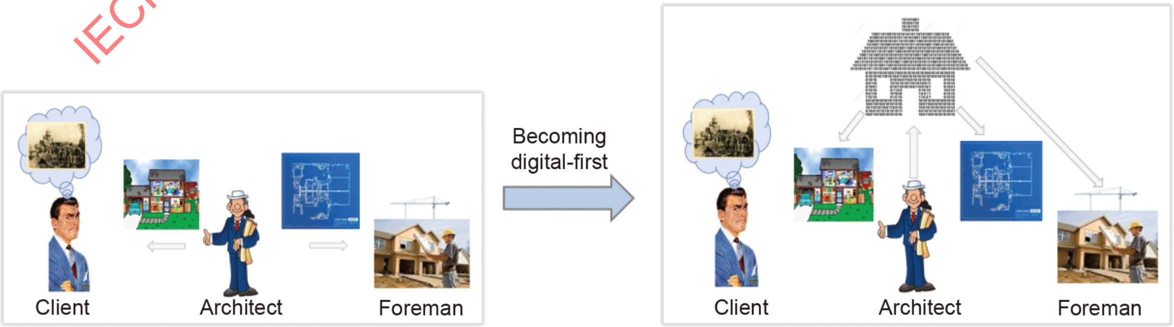
Figure 8 – A dependency matrix "Strategic goals" and "Solution space constraints" versus "Essential desirable characteristics"

5.19 Architecture principles of the solution space

5.19.1 Digital orientation

Right now, many material objects (house, city, human, etc.) and immaterial objects (music, project plan, etc.) may have a digital representation (3.4.2) of the object (3.3.15). The two expressions "a digital representation of an object" and "a digital object" are often used as synonyms. Because a few presentations of the same object may coexist at the same time, it is necessary to define explicitly which presentation is primary (or source of truth) and which presentation(s) is(are) secondary.

For a natural object, its digital representation is always secondary. For example, the digital representation of a person is always secondary. For a man-made object, its digital representation may be primary. For example, a house may be designed fully digitally before being built. See Figure 9.



IEC

Figure 9 – Becoming digital-first

However, the dependencies between various presentations may be rather complex and primacy may be dynamic. Figure 10 shows the built house (physical) which is equipped with various sensors (think about "Internet of Things") to provide another digital representation, i.e. the built house (digital). The latter can be compared with the house design (digital representation) to find out some deficiencies of the built house (physical representation). Also, the built house (digital representation) can use a variety of data modelling techniques to assess various likely impacts in order to optimize the house design (digital representation) and the house built (physical representation).

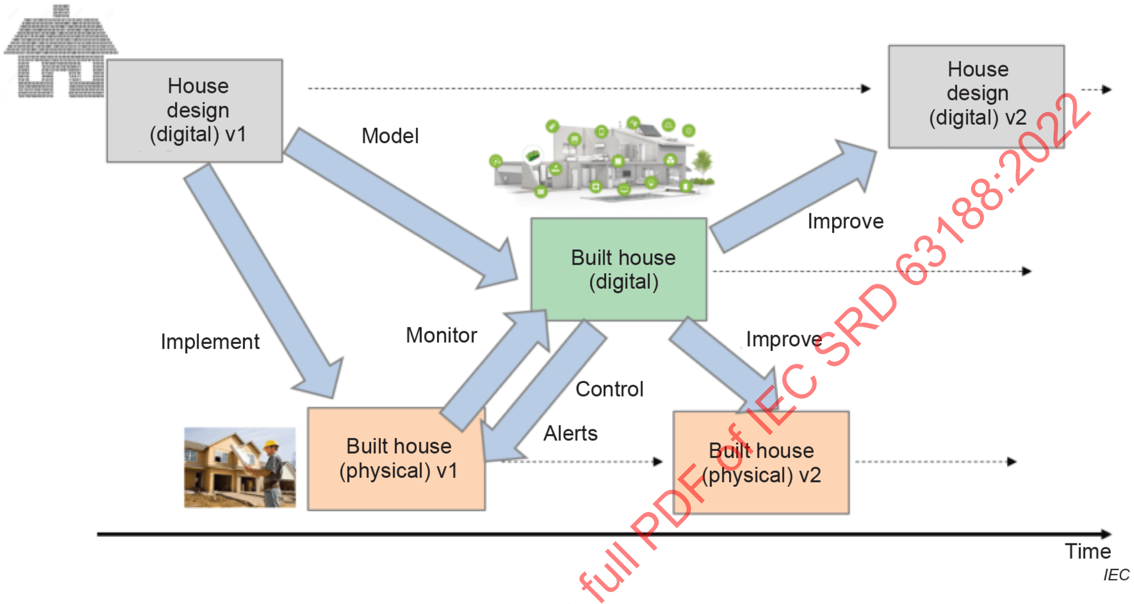

House design (digital) v1
Model
House design (digital) v2
Improve
Built house (digital)
Implement
Monitor
Improve
Control
Alerts
Built house (physical) v1
Built house (physical) v2
Time
IEC

Figure 10 – Many representations of the same house and relationships between them

Any complex entity becomes a digitally coordinated system (3.4.1). A region, an economy, a university, an enterprise, a city, and a hospital are examples of potential digital systems.

- Digital economy is an economy which is built as a digitally coordinated system.
- Digital enterprise is an enterprise which is built as a digitally coordinated system.

All supporting tools, assets, etc. within and beyond a digital system also could become digital as well. The following are examples.

a) Sovereign currency (only the physical cash or money "M0") gets its digital representation, e.g. central bank digital currency, to move it directly from one owner to another.

b) Enterprise shares become digital tokens.

c) Laws become digital (i.e. formal, explicit, computer-readable and computer-executable) and act objectively.

d) Rights of ownership on an asset become digital records which are maintained in digital archives and managed by its owner directly; assets may be digital or material; whole or partial.

e) Contracts become digital contracts.

f) Documents become structured, machine-readable and machine-executable.

g) Identification of a person becomes digital and online.

h) Business processes become digital processes.

i) Horizontal and vertical integration between organizations are carried out by inter-organizational digital processes.

j) Terminology becomes ontology (but it may have a few intermediate steps such as glossary and system of concepts).

Rebuilding existing systems as digitally coordinated systems is about digital transformation (3.4.3). The main benefit of DT is that digital elements of digital systems can be very easily repeated or cloned (a digital element is not a material object like one kilometre of a highway or a car). Therefore, if digital systems are created from, mainly, digital repeatable elements then the cost of modelling and prototyping DT decreases, the time of DT reduces, and the quality of DT increases.

The importance of the digital orientation for smart cities is based on the following logic.

2) Cities, as places where people live and work, are very important for our civilization as global populations in urban areas are projected to grow whilst access to resources is finite. Careful management of our cities using smart technologies is key to sustain them.

3) All cities are different but many share common characteristics, issues and needs.

4) Smart city is a city which makes the world better for its citizen (3.2.1), business and city administration (3.2.3).

5) In smart cities projects, 50 % to 70 % of work is about IT[3] thus digital.

6) Smart city is a city which is (re)built as a digital system (DT is critical).

7) Digital systems could be implemented well, be adaptable and easy to repeat, but such desired characteristics shall be purposefully created.

8) If a common methodology for building smart cities as digital repeatable systems could be used, then all smart cities share similar issues, needs, capabilities and components.

9) Coordinated and cooperated building of smart cities as digital repeatable systems could lead to substantial gains in cost reduction, speed of execution and increased quality of design and construction.

10) An export version of the smart cities implementation that can be shared with other communities could be created automatically.

11) Consistent design, practices and components of smart cities can apply for healthcare, buildings, homes, industries, territories and governmental agencies enabling procurement savings and greater access to fine grained data for greater insights and improved decision making.

5.19.2 Platform-enabled agile solutions orientation

The logic behind the platform-enabled agile solutions is rather straightforward – instead of developing traditional applications which contain a lot of duplicating functionality, all common functionality is collected into a platform (3.3.17) and solutions which are built on top of that platform possess only unique functionality. See Figure 11.

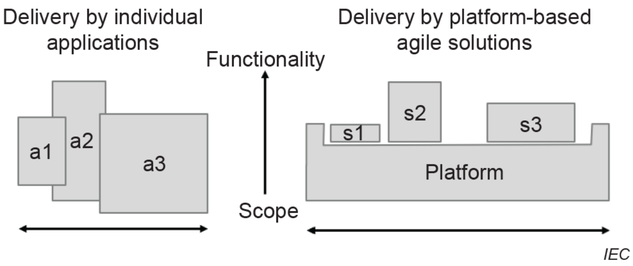

Delivery by individual applications
Delivery by platform-based agile solutions
Functionality
Scope
a1
a2
a3
s1
s2
s3
Platform
IEC

Figure 11 – Platform-based agile solutions pattern

[3] This estimation was given at a conference in Varanasi in December 2018 about the Indian programme "100 Smart Cities".

Platforms should be designed on the following principles.

a) The platform simplifies the use of its platform components and ensures interaction between them. Thus, the platform frees up resources to focus on solving unique problems.

b) New solutions are implemented outside the platform using rapid development techniques (agile methodology) and the platform’s existing functionality.

c) The planning of new solutions is coordinated within the scope of the platform to minimize duplication of efforts in solving the same problems.

d) Successful new (thus innovative) solutions are gradually included in the platform for their wide use.

e) Constant improvement of the platform is achieved through transparency, feedback, results of use and systematic evaluation of existing platform components.

f) All interactions inside and outside the platform use the standardized interfaces (API methodology).

g) The platform provides for several interchangeable options for some platform components.

h) The platform offers not only a rich functionality, but also a methodology for developing various solutions based on it.

i) The platform is modular by design, thus reducing complexity and cost when increasing its functionality horizontally and vertically.

The platform-enabled agile solutions orientation works well with the digital orientation – all the functionality common among smart cities forms a platform and any unique functionality (from a particular smart city) is implemented in the solutions on top of the platform. The logic (see also Figure 12, Figure 13 Figure 14 and Figure 15) is the following.

1) Create a common understanding about smart cities.

2) Adopt a systemic approach to isolate and commoditize common functionality.

3) Support various integration methods for both unique parts and common parts of smart city systems.

4) Cooperate to develop and share repeatable digital solutions which are acceptable for Common Smart Cities Digital Repeatable Platforms.

5) Create an individual version of the Common Smart Cities Digital Repeatable Platform for each smart city.

6) Provide a framework to help a smart city tailor the platform for its unique needs.

7) Collaborate to deliver better outcomes of higher quality, delivered in less time and for less money.

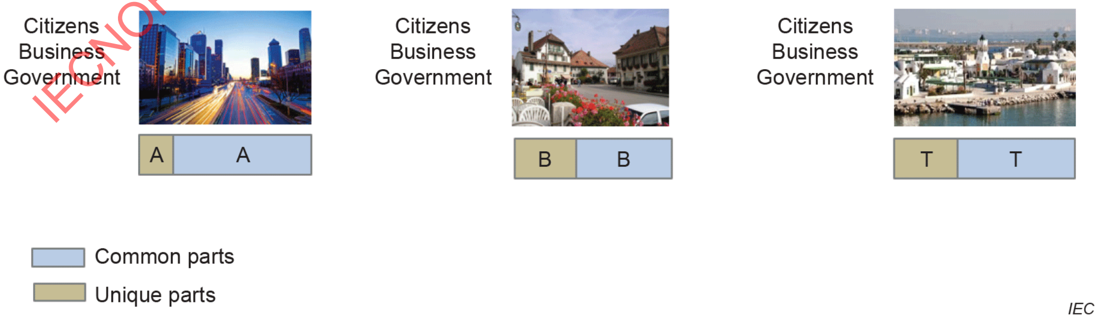


Figure 12 – Unique and common functionality in each city

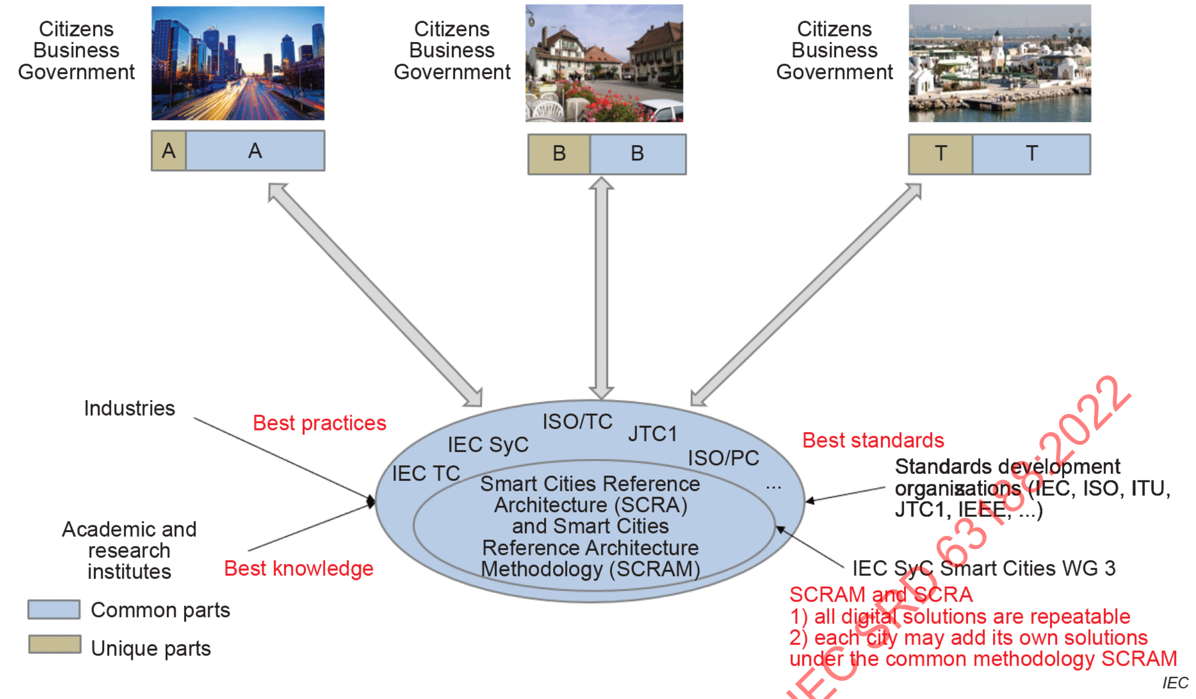

Citizens
Business
Government
A
A
Citizens
Business
Government
B
B
Citizens
Business
Government
T
T
Industries
Best practices
ISO/TC
JTC1
IEC SyC
ISO/PC
IEC TC
...
Smart Cities Reference Architecture (SCRA) and Smart Cities Reference Architecture Methodology (SCRAM)
Best standards
Standards development organizations (IEC, ISO, ITU, JTC1, IEEE, ...)
Academic and research institutes
Best knowledge
IEC SyC Smart Cities WG 3
SCRAM and SCRA
1) all digital solutions are repeatable
2) each city may add its own solutions under the common methodology SCRAM
Common parts
Unique parts
IEC

Figure 13 – Reference architecture helps agreement on common functionality of Smart Cities

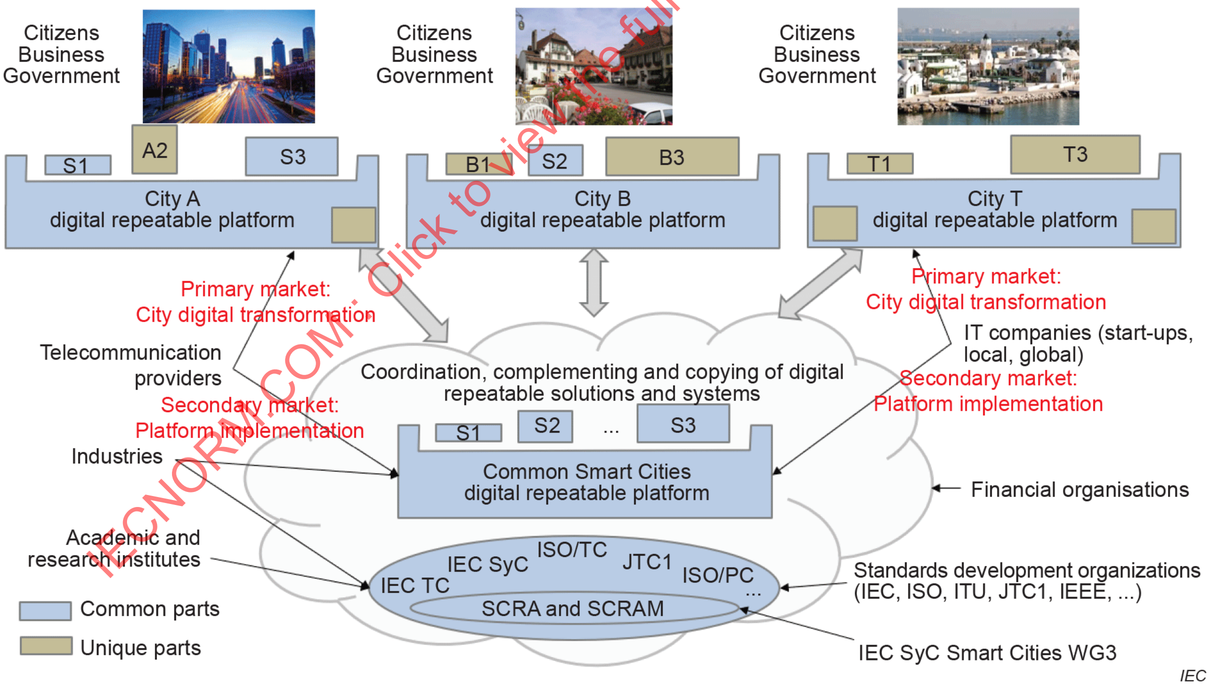

Citizens
Business
Government
A2
S1
S3
City A
digital repeatable platform
Citizens
Business
Government
B1
S2
B3
City B
digital repeatable platform
Citizens
Business
Government
T1
T3
City T
digital repeatable platform
Primary market:
City digital transformation
Telecommunication providers
Secondary market:
Platform implementation
Industries
Academic and research institutes
Coordination, complementing and copying of digital repeatable solutions and systems
S1
S2
...
S3
Common Smart Cities
digital repeatable platform
Primary market:
City digital transformation
IT companies (start-ups, local, global)
Secondary market:
Platform implementation
Financial organisations
ISO/TC
IEC SyC
JTC1
IEC TC
ISO/PC
...
SCRA and SCRAM
Standards development organizations (IEC, ISO, ITU, JTC1, IEEE, ...)
IEC SyC Smart Cities WG3
Common parts
Unique parts
IEC

Figure 14 – Common Smart Cities Digital Repeatable Platform

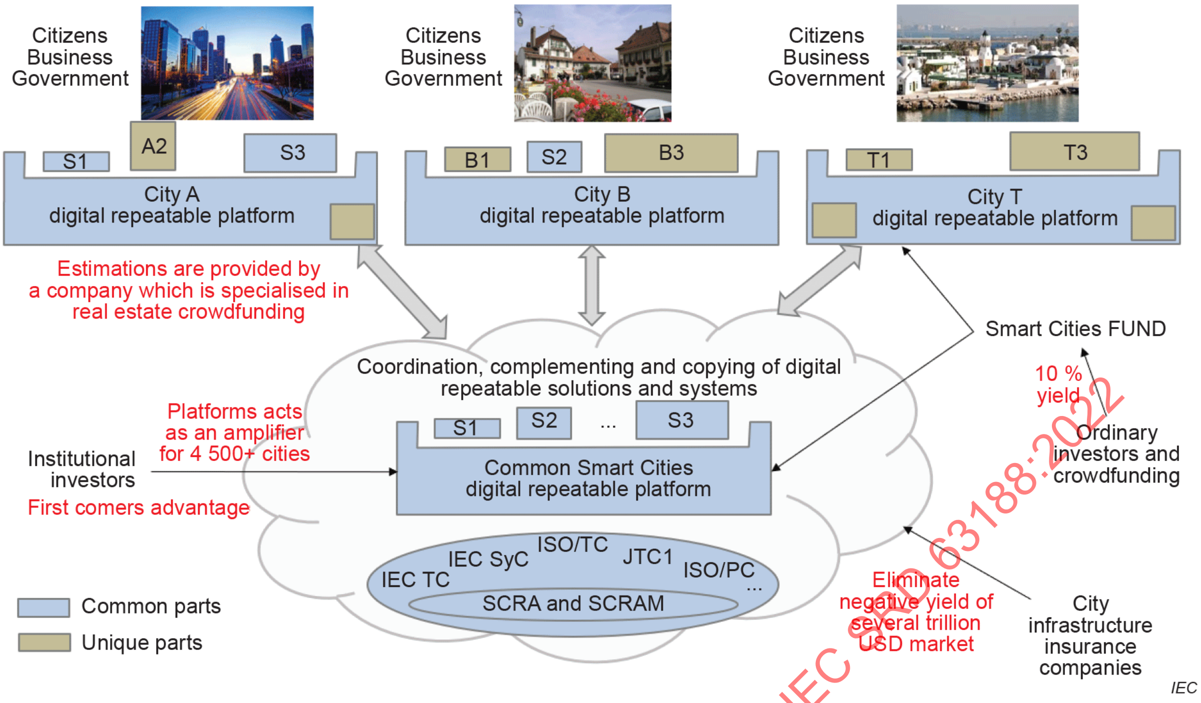

Citizens
Business
Government
A2
S1
S3
City A
digital repeatable platform
Citizens
Business
Government
B1
S2
B3
City B
digital repeatable platform
Citizens
Business
Government
T1
T3
City T
digital repeatable platform
Estimations are provided by a company which is specialised in real estate crowdfunding
Coordination, complementing and copying of digital repeatable solutions and systems
Platforms acts as an amplifier for 4 500+ cities
Institutional investors
First comers advantage
S1
S2
...
S3
Common Smart Cities
digital repeatable platform
Smart Cities FUND
10 % yield
Ordinary investors and crowdfunding
IEC TC
IEC SyC
ISO/TC
JTC1
ISO/PC
...
SCRA and SCRAM
Eliminate negative yield of several trillion USD market
City infrastructure insurance companies
Common parts
Unique parts
IEC

Figure 15 – Investment opportunities

A few simple calculations below show a potential effect of standardization for implementation of smart cities. N is the total cost of a smart city construction only, of which **70 %** is common and **30 %** is unique. Thus, the total cost for 100 smart cities **without standardization** is N × 100. At the same time, the total cost for 100 smart cities **with standardization** is N × 100 × 0,3 (unique parts) + N × 1 × 0,7 (common parts) × 3 (complexity factor) = N × (30 + 2,1) = N × **32,1**. So, the cost difference is (N × 100) / (N × 32,1), which is about **3 times**! Also, there will be some gains in quality and time.

The complexity of smart cities will require nested layers, with each of them acting as a platform. A Common Smart Cities Digital Repeatable Platform may have the several layers (universal, urban and some urban-specific). See Figure 16.

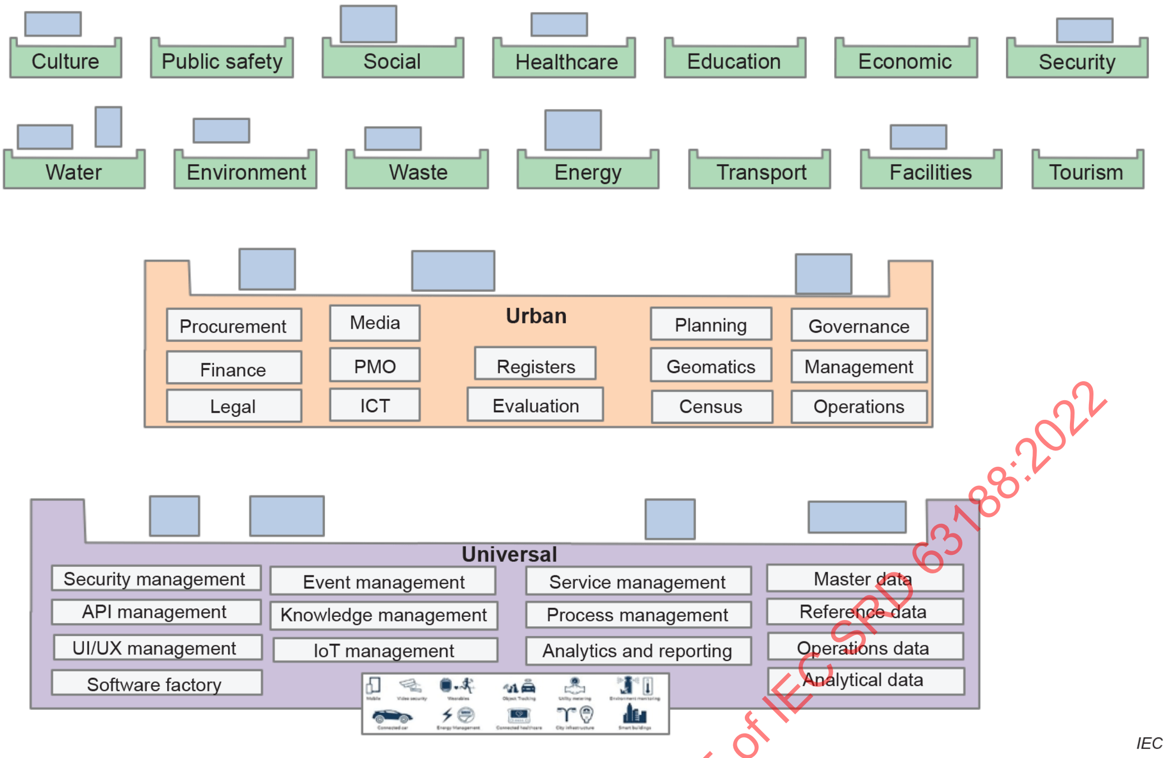


Figure 16 – An example of Common Smart Cities Digital Repeatable Platform structure

Ideally, platform components are off-the-shelf products for particular technologies.

The platform orientation does not imply that all the smart cities will use the same copy (installation) of the Common Smart Cities Digital Repeatable Platform. On the contrary, each smart city will have its own copy (installation) of the Common Smart Cities Digital Repeatable Platform, which will evolve with the city pace and address the city's unique needs.

The platform orientation does not need a perfect version of the Common Smart Cities Digital Repeatable Platform to be built up-front. The Common Smart Cities Digital Repeatable Platform will be built incrementally by starting from a Minimum Viable Product (MVP) version.

5.19.3 Capabilities, functions and processes orientation

Capability is an ability of a system or a system element to do something at a particular level of performance. Capability is an intrinsic characteristic of a system or a system element. A capability is simultaneously a) unit-of-functionality (which contributes to fulfilment of the system mission) and b) unit-of-performance (which is governed by the system vision). Any system or a system element shall possess one or more capabilities. Capability is more abstract than processes, functions, organization chart, etc.

In architecting and engineering of systems, a typical working pattern is the following:

- a target system shall possess a requested capability (requested functionality and requested performance);
- the architected and engineered system has an estimated capability (estimated functionality and estimated performance);
- the implemented system exhibits a demonstrated capability (demonstrated functionality and demonstrated performance);
- if there are unacceptable gaps between requested functionality and demonstrated functionality or requested performance and demonstrated performance then the system may be reengineered or abandoned (i.e. considered impossible to implement).

So, there are a minimum two interpretations of capability – demand (i.e. requested) and supply (i.e. demonstrated). If there are a few steps in the implementation (e.g. requirements – architecture – implementation) then a few demand–supply pairs can be used.

If a capability is "big" then it may be decomposed into "smaller" capabilities. Although the decomposition of functionality is rather straightforward, the decomposition of performance is often tricky. In complex decomposition cases, the best (so far) way to assure that requested, expected and demonstrated performances are well-matched is to consider a "bigger" capability as a digital process which explicitly coordinates the "smaller" capabilities. Thus, it would be possible: a) to simulate the "bigger" capability to understand the estimated performance, and b) to use the digital process as an integral part of the implementation (therefore minimize differences between architecting and engineering, and implementation.

There are three facets of performance which are associated with a capability:

1) requested by users of this capability;
2) estimated during its architecting and engineering; and
3) demonstrated during its operations.

In some case, a simple variant of the concept "capability" is used, it is the concept "reference capability" which disregards performance. Reference capability models of some industries are based on reference capabilities. Decomposition of "big" reference capabilities into "smaller" reference capabilities may be expressed by a simple list.

Because any implementation means some choices between cost, risks, quality, importance, etc. it is necessary to take them into account (usually known as sourcing policies "buy", "build", rent", etc.) and consider the costs and benefit in relation to short-, medium- and long-term costs and benefits with analysis of the cost to transfer from one approach or strategy to another. For example, some capabilities are commodities which can be obtained from the market. There are several options for such sourcing:

- a capability will be implemented within a system (as a new function);
- a capability will be purchased from an external company as a product;
- a capability will be purchased from an external company as an acquisition;
- a capability is already available within a system (as a shared function);
- a capability is sourced from a partner;
- a capability is sourced from commodity markets; and
- a capability is ignored.

Function of a system or a system element is the capability of the system or the system element which is implemented within the system or the system element.

The following recursive procedure may be used for defining functions.

a) Select a capability.
b) Take a decision on how to source it.
c) If this capability is a new function and it is "big", then provide a description of the function including its implementation (in accordance with IDEF0, ISO/IEC 19510:2013 [4] [23] or other techniques) via "smaller" capabilities.
d) Proceed, if necessary, to a).

[4] ISO/IEC 19510:2013 is identical to OMG BPMN™ 2.0.1. BPMN is a trademark of Object Management Group (OMG). This information is given for the convenience of users of this document and does not constitute an endorsement by IEC.

The results of this procedure are: capability map, process map, function map, partners nomenclature, products nomenclature and relationships between them.

5.19.4 Services, microservices and APIs orientation

Having capabilities is not enough to serve them to an external client. Service of a system or a system element is a group of one or more capabilities of the system or the system element accessible using a prescribed interface. In a digital system, an interface can be

a) Machine-to-Machine (M2M),

b) software capability-to-software capability, also called Application Programming Interface (API), and

c) human-to-machine, also called User Interface (UI).

Also, in digital systems there are two additional (to unit-of-functionality) concepts:

- unit-of-deployment – group of one or more units-of-functionality which, individually and independently of any other such group, can be deployed intact to a digital computer or a digital system; and
- unit-of-execution – group of one or more units-of-deployment which can be executed within a digital computing process.

These two concepts allow to define microservice as a service comprising a single unit-of-functionality packaged exclusively in a single unit-of-deployment to form a single unit-of-execution. Microservices are designed to implement some elements of a system as a set of microservices:

- an application (unit-of-deployment that interacts with its users by provisioning them with several units-of-functionality via a human–computer interface); or
- a solution (a scaled-down application in the "platform-based agile solutions" paradigm).

When a set of microservices implements an application or a solution, such microservices are

- dependent at the design-time because they are engineered together,
- independent at the deployment-time because they are autonomous (at some extent), and
- interdependent at the run-time, because they contribute to a common goal.

Microservices have been introduced in [24] and became very popular for achieving agility, reliability and scalability of applications and solutions. There are many, sometimes extreme, recommendations about microservices. For example, some recommendations require existing "monoliths" (single units-of-deployment with hundreds of units-of-functionality) to be converted to microservices.

Knowing that microservices are well-suited as digital repeatable elements, for the purposes of the Common Smart Cities Digital Repeatable Platform the following microservices-related rules are applied.

1) If economically reasonable then some units-of-functionality may be provisioned from monoliths (typically, off-the-shelf products).

2) A service or microservice may invoke other microservices, services and monoliths.

3) If possible, services are implemented to be easily decomposed into microservices and vice-versa, several microservices should be easily composed into a single service.

4) All interactions between units-of-functionality shall be carried out via API, expressed in a few standard IDLs, e.g. OpenAPI 3.0 [25] and ISO/IEC 19516:2020 [26], 4.2.

See Figure 17.

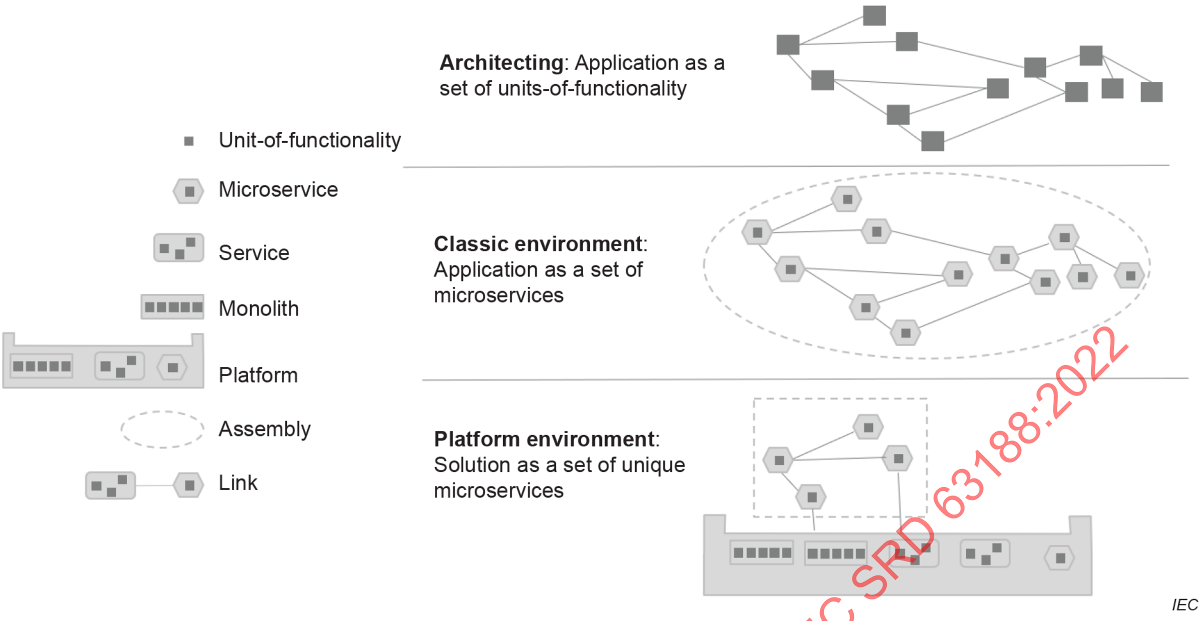

Architecting: Application as a set of units-of-functionality
Unit-of-functionality
Microservice
Service
Monolith
Platform
Assembly
Link
Classic environment: Application as a set of microservices
Platform environment: Solution as a set of unique microservices
IEC

Figure 17 – Using together services, microservices and monoliths

Often, microservices are blamed for increasing complexity by allowing any microservice to be linked to others. Such a drawback can be avoided by using digital processes. See Figure 18.

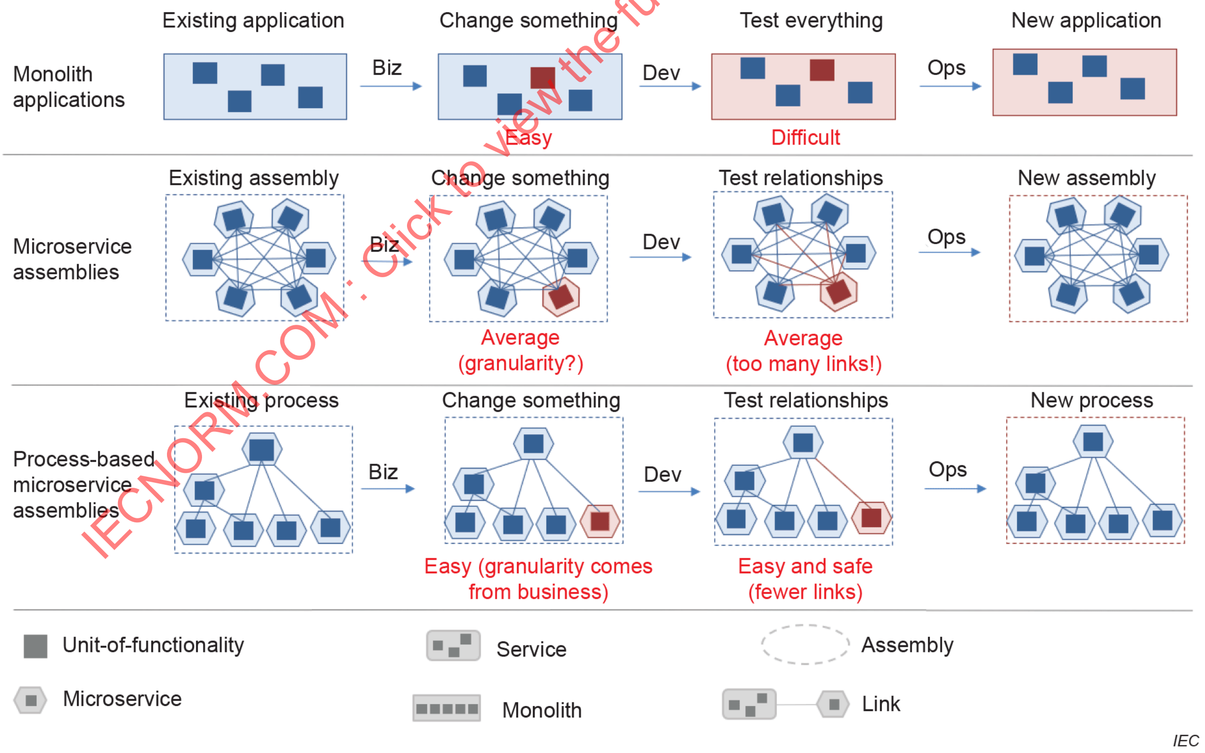


Figure 18 – Advantage of process-based microservice assemblies

5.19.5 Common engineering practices orientation

The platform-enabled agile solutions orientation implies common engineering practices for solutions created on top of the Common Smart Cities Digital Repeatable Platform. Any solution which is built on such a platform shall contain only functionality which is not implemented in the platform. To achieve this, decomposition of a solution shall be done not in terms of software artefacts such as classes, packages, interfaces, schemas, etc., but in terms of solution artefacts such as:

- events,
- processes,
- UI forms,
- roles,
- rules,
- KPIs,
- audit-trails,
- reports,
- activities,
- automation procedures,
- data structures,
- documents,
- IoT,
- etc.

To facilitate isolation of such artefacts, a reference application architecture for solutions is proposed (see Figure 19).

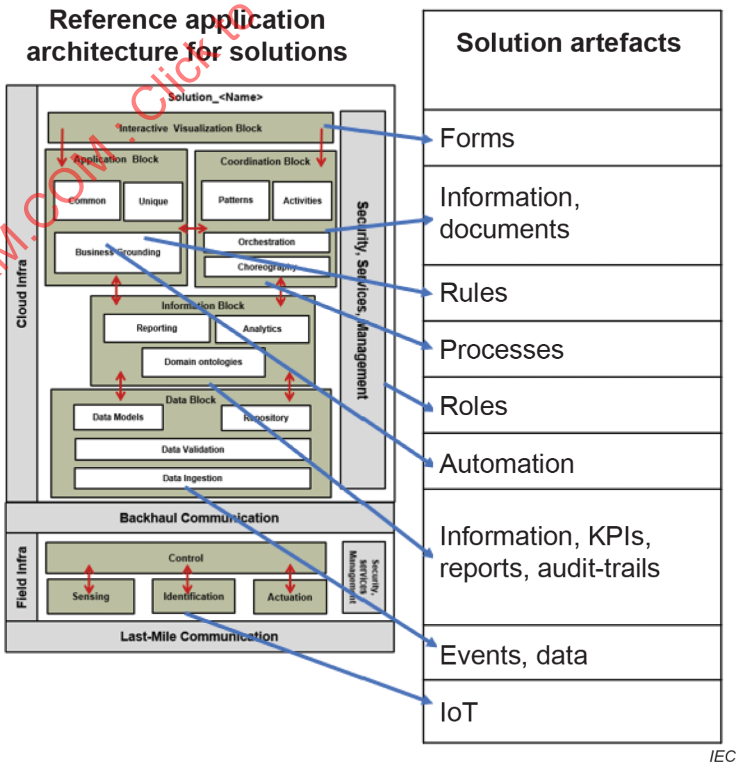


Figure 19 – Reference application architecture for solutions

However, solution artefacts may have several facets:

a) specific management tool, e.g. Business Process Management (BPM)-suite tool;

b) templates, e.g. process templates;

c) instances, e.g. process instances;

d) APIs (or interfaces) to all functionality;

e) patterns used by templates, e.g. workflow patterns.

Naturally, specific management tools, APIs, patterns are needed only once for the whole city system, thus they shall be in the platform. However, templates and instances usually belong to solutions. See Figure 20 with mapping of process artefact facets to platforms and solutions.

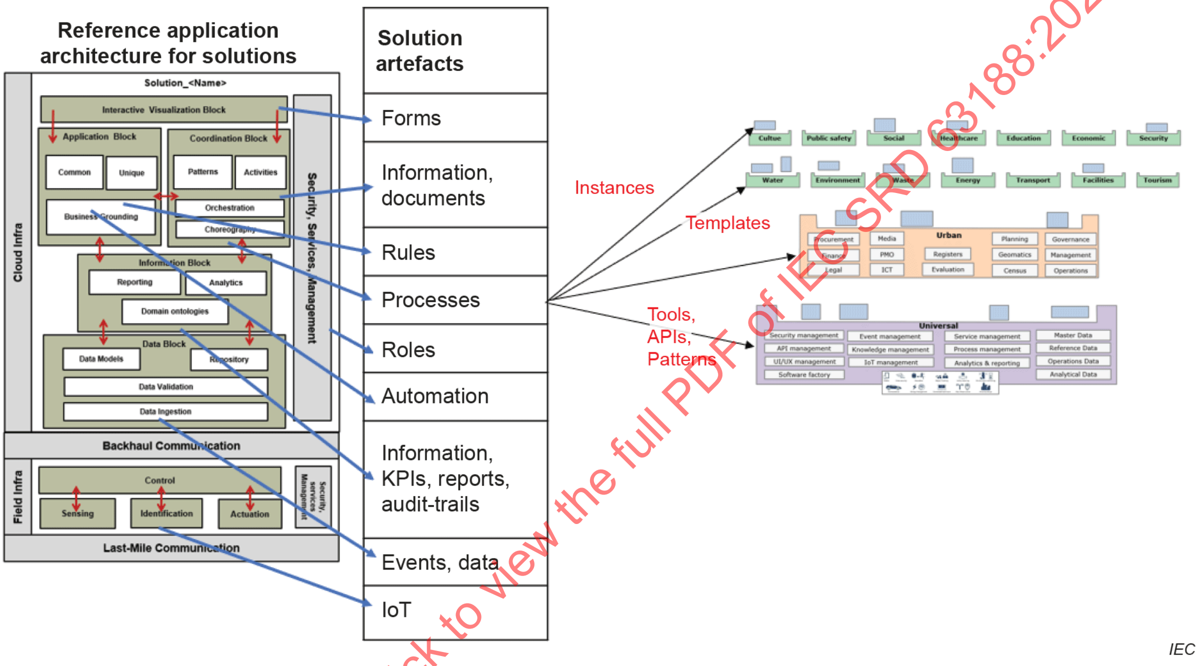


Figure 20 – Mapping various artefact facets to platforms and solutions (example)

The reference application architecture for solutions will be further elaborated in the SCIM by adding a typology (classification) of solutions. This means that there are some types of solution, and each type is formed around a predominant solution artefact, namely:

– event centric,

– data-entry centric,

– document/content centric,

– data or/and information flow centric,

– data or/and information visualization,

– IoT-device centric,

– short-running operations (activities-based),

– long-running operations (processes-based),

– etc.

However, a real solution, usually, combines some of those types.

Thus, solutions are delivered in accordance with the following generic procedure:

1) minimal architecting to understand the type of solution;
2) collecting use cases to define capabilities of the solution;
3) quick prototyping to outline functions, APIs, and other solutions artefacts;
4) gap analysis to determine what is missing in the platform;
5) developing missing functionality as microservices to close the gap;
6) assembling to deliver the solution.

5.19.6 Crosscutting aspects orientation

Various non-functional or quality characteristics of systems are actually dependent on all the system elements and relationships between them. For example, the security design "weakest link" principle says that in designing security for a system, focus should be put on the weakest components of the overall system [27].

The understanding of the emergent nature of various non-functional and quality characteristics led to raising of requirements for guaranteeing such characteristics "by-design". The most noticeable example of "privacy by-design" is the General Data Protection Regulations (GDPR) [28] from the European Union. The concept "entity characteristic by-design" may be defined as taking into account the characteristic throughout the entity entire lifecycle processes (e.g. architecting, engineering, construction, operating, etc.) as an essential emergent characteristic of the entity.

For other characteristics by design, it is necessary to "spread" such characteristics over the system elements and relationships between them. For example, see Figure 21, knowing the security-related paraments for each least granular system element (oval "Security") and having relationships between the system elements (oval "Enterprise architecture"), it is possible to objectively evaluate system risks (oval "Risks"). After reaching the proper level of granularity, the technique "Unified Security Model" from ITU SG17-C758 [29] document can be used.

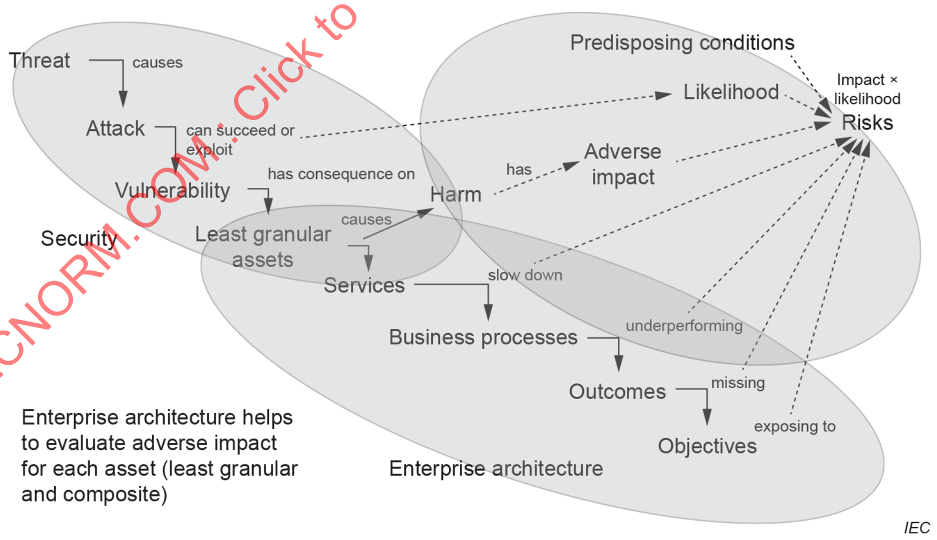


Figure 21 – Linking security, architecture and risks

The same can be done for privacy, safety, reliability and resilience, and other emergent characteristics. Considering that relationships are static and dynamic, the structure and behaviour of such emergent characteristics will be covered.

For each system element, there are internal and external crosscutting aspect enhancement practices. The crosscutting aspect enhancement internal practices comprise various certifications and industry agreements which are aspect-specific and element-specific. The crosscutting aspect enhancement external practices comprise the treatment of an element as:

- "white-box" with full internal visibility and testability – some simulation and compliance crosscutting aspect enhancement external methods can be used;
- "black-box" with no internal visibility and testability – use crosscutting aspect enhancement external points for data/information inputs and outputs, and crosscutting aspect enhancement external controls for leaks detection;
- "grey-box" with partial internal visibility and testability – combining two previous options.;

Typically, crosscutting aspect enhancement external methods are based on digital processes. Crosscutting aspect enhancement external points are based on digital archives and technologies similar to Security Information and Event Management (SIEM) tools. Crosscutting aspect enhancement external controls are monitoring tools with some Artificial Intelligence (AI) logic, for example, Data Loss Protection (DLP) tools.

For each relationship between elements, use crosscutting aspect enhancement external points for data/information flow starts and ends, and crosscutting aspect enhancement transit controls.

The static relationships are expressed by any connection between elements, but dynamic relationships are expressed only by events (EDA and EPN techniques), processes (BPM techniques) and information flows (IFD techniques).

Thus, an explicit description of system elements and relationships between them provides a nomenclature of crosscutting aspects practices, controls, points, and methods to be added to the system. Then they shall be linked to the risk management practices.

5.19.7 Common governance, management and operations practices orientation

Because of the "platform-enabled agile solutions" orientation, smart cities should apply (as much as possible) common governance, management and operations practices. This will allow the building of a common knowledge base and streamline improvements. Examples are system lifecycle procedures, solutions artefacts, etc.

As the building of such practices has incremental nature, they will be addressed in the SCIM.

5.19.8 Existing knowledge orientation

Because of the complexity of city systems, the SCRAM and SCRA use, if applicable, existing bodies of knowledge, industry and international standards. Obviously, the "integration" of them into city systems is a long process (for example, even terminology is, very often, different), thus such "integration" will be addressed in the SCIM.

At present, the areas of potential use in the SCRAM of a particular body of knowledge or standards are explicitly indicated. To preserve the conceptual integrity of the SCRAM and SCRA, these definitions, models, artefacts and decisions take precedence over other bodies of knowledge and standards. For example, if the same artefact-type or an artefact is used in SCRAM and another body of knowledge, the SCRAM artefact-type shall be used and linked.

5.20 Dependency matrix "Essential desired characteristic" versus "Architecture principles"

See Figure 22.

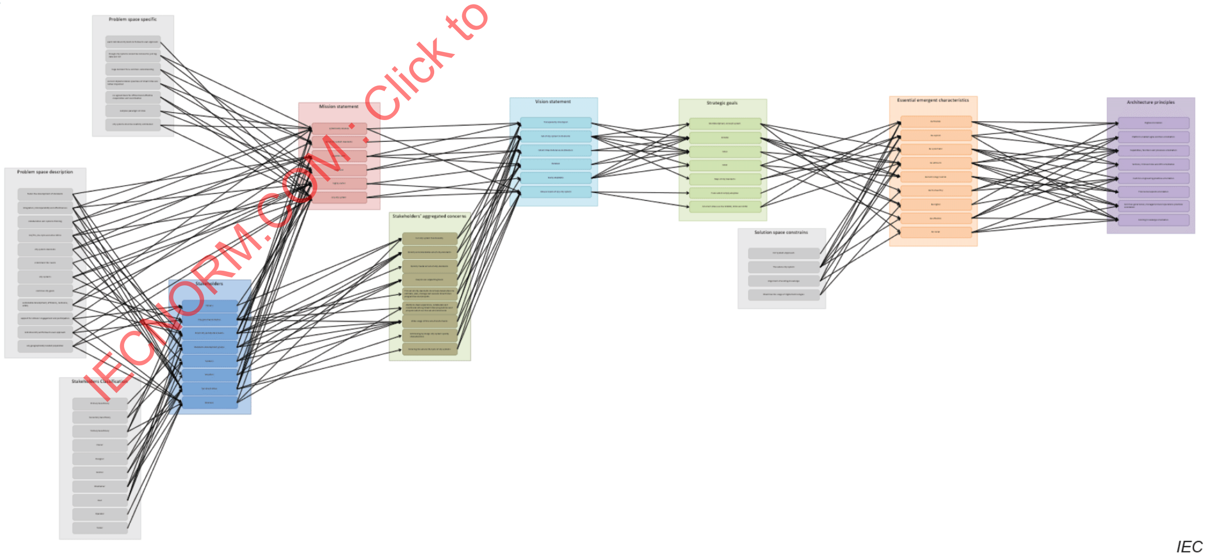
ROWS are sources (Essential emergent characteristics) vs. COLUMNS are destinations (Architecture principles)	Digital orientation	Platform-enabled agile solutions orientation	Capabilities, functions and processes orientation	Services, microservices and APIs orientation	Common engineering practices orientation	Transversal aspects orientation	Common governance, management and operations practices orientation	Existing knowledge orientation
Be flexible	X		X	X				
Be explicit	X		X					X
Be systematic				X	X	X	X	
Be efficient		X	X			X	X	X
Be technology neutral						X	X	X
Be trustworthy		X	X		X			
Be digital	X				X			
Be effective		X	X	X				X
Be social					X		X	X

IEC

Figure 22 – A dependency matrix "Essential desired characteristic" versus "Architecture principles"

5.21 All dependencies together

The full set of relationships is depicted in Figure 23.



IEC

Figure 23 – All relationships

It is possible to highlight only relationships for a particular element. See Figure 24.

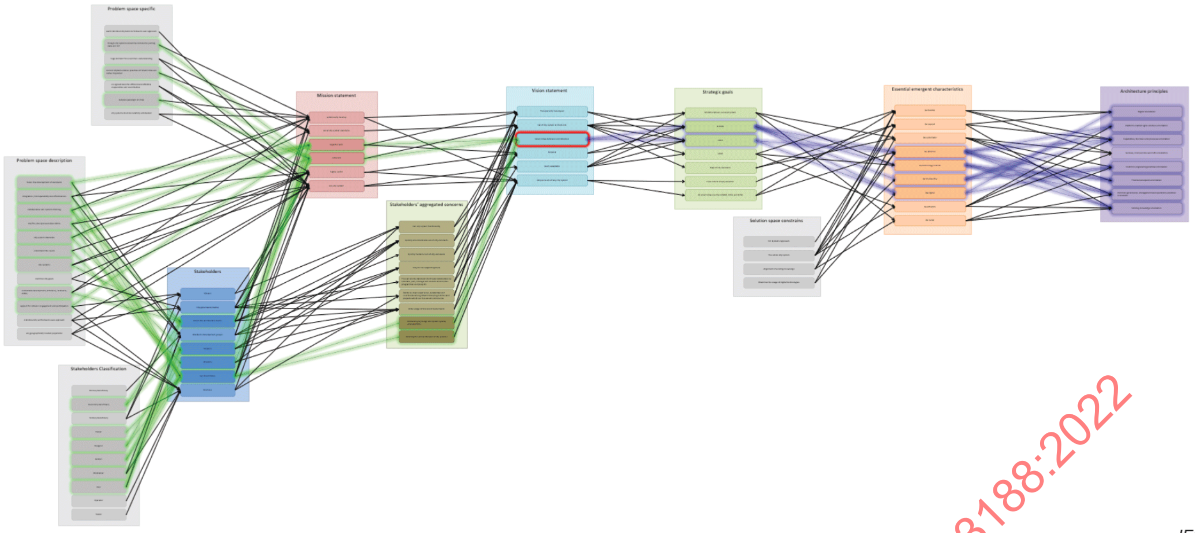

Problem space specific
Mission statement
Vision statement
Strategic goals
Essential emergent characteristics
Architecture principles
Problem space description
Stakeholders' aggregated concerns
Solution space constraints
Stakeholders
Stakeholders Classification
IEC

Figure 24 – All relationships with highlighted ones (see the attached file for details)

6 Outline of the SCRAM

6.1 SCRAM data schema

The SCRAM is a set of SCRAM viewpoints (see clause 7), SCRAM model-types (see clause 8) and SCRAM artefact-types (see Annex D). Each SCRAM viewpoint comprises one or more SCRAM model-types. Any SCRAM model-type should be in one or more SCRAM viewpoints. Any SCRAM model-type is a relationship between various SCRAM model-types and SCRAM artefact-types. There are three variants of such a relationship:

- between some SCRAM model-types;
- between some SCRAM model-types and some SCRAM artefact-types;
- between some SCRAM artefact-types.

All the SCRAM viewpoints, SCRAM model-types and SCRAM artefact-types are independent from any smart cities system. Other viewpoints, model-types and artefact-types may be added to extend the SCRAM for purposes of a particular smart city. Annex A provides viewpoints and model-kinds from the Zachman Framework™ [30] which can be added if required.

SCRAM model-types are typically maps (spatial/graphical arrangements of map elements), nomenclatures (list-like arrangement of map elements), diagrams (arrangements of map elements in accordance with some notations), semi-formal descriptions and all popular modelling notations. Semi-formal descriptions are free texts with highlighted keywords which can be referenced in other model-types. See examples in Clause 5.

Every SCRA shall conform to a SCRAM viewpoint. Every SCRA model shall confirm a SCRAM model-type. Every SCRA artefact shall confirm to a SCRAM artefact-type. All the SCRA views, models and artefacts are dependent on a particular smart city (or system-of-interest in Figure 25).

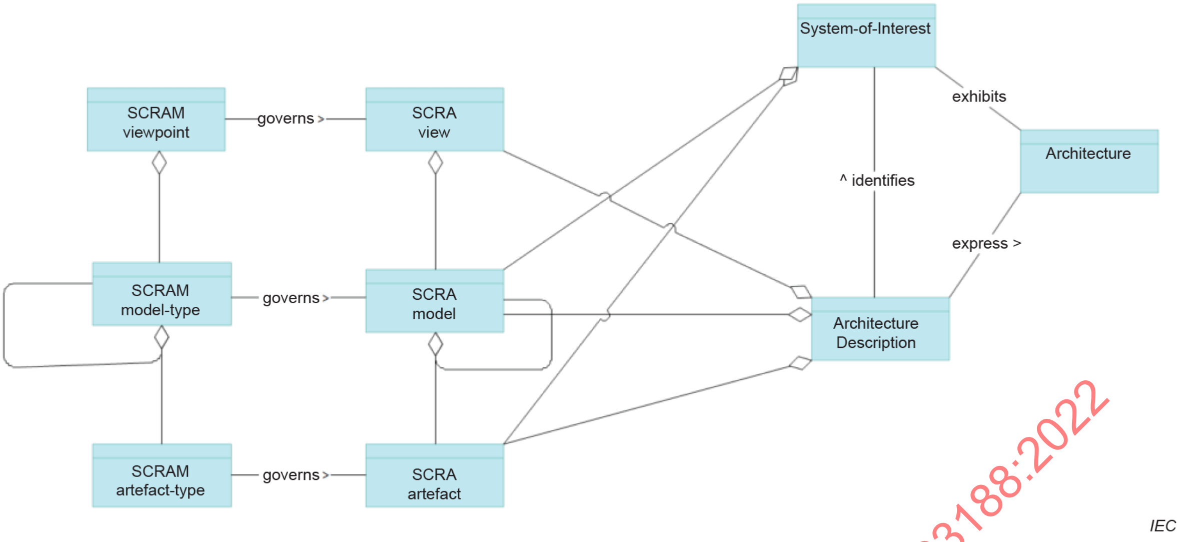


Figure 25 – Fragment of the architecture elements relation

The architecture viewpoint and architecture view "justification" (which is introduced in 5.1) is not yet a SCRAM viewpoint or a SCRA viewpoint.

6.2 How the SCRA will be created by the SCRAM

The SCRA (which is an architecture of an imaginary smart city) will be developed via the SCRAM in the following generic algorithm:

a) select a SCRAM viewpoint from the SCRAM and define a SCRA view for the SCRA;

b) select a SCRAM model-type from this SCRAM viewpoint;

c) construct a SCRA model (for the imaginary smart city) in accordance with this SCRAM model-type;

d) define all necessary SCRA artefacts for this SCRA model in accordance with their SCRAM artefact-types; and

e) continue to item b) if there are more SCRA models in the SCRA view to be constructed;

f) continue to item a) if there are more SCRA views in the SCRA to be constructed.

There is a natural order between the SCRAM viewpoints and SCRAM model-types as shown in Figure 26. However, this order is not the rigid sequence of applying the SCRAM, because the SCRAM is built under consideration that some SCRA models may be changed in an arbitrary order as a normal situation with architecting smart cities systems. For example, during the process of developing a solution, some extra information about a problem may be gained and that may lead to a necessity for changing some existing SCRA models. It is mandatory to validate after any changes that all the SCRA models and artefacts continue to be aligned.

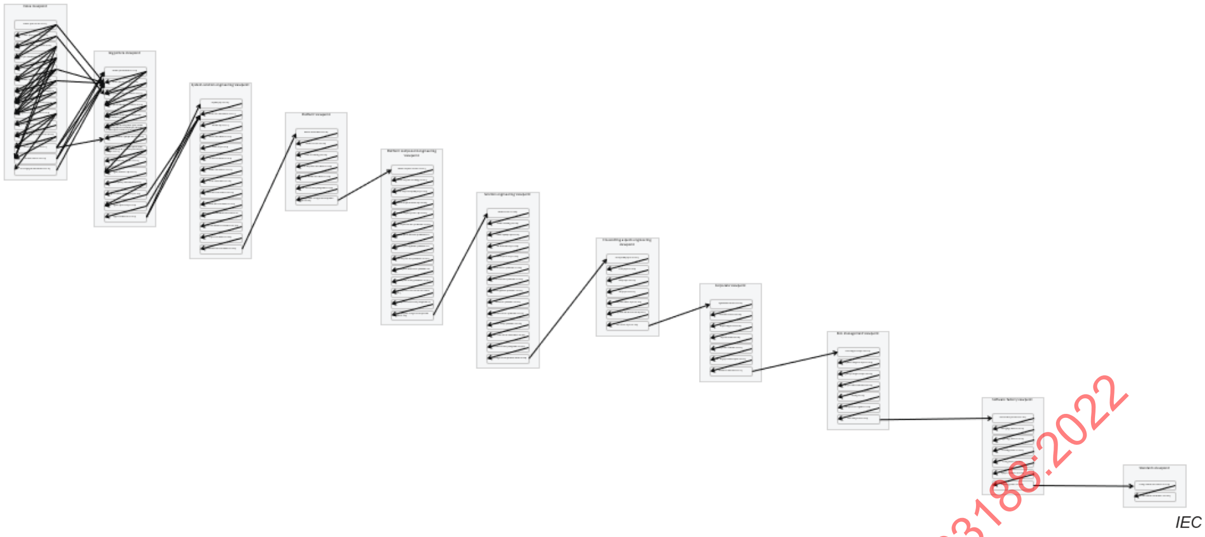


Figure 26 – Dependencies between the SCRAM model-types

Because the SCRA is an architecture of a system-of-systems, the generic algorithm may be applied several times – once for the whole city and then for each vertical such as water management, waste management, energy management, transport management, etc.

6.3 Tailoring the SCRAM and SCRA

The general approach for tailoring the SCRAM and SCRA is rather straightforward – extend existing and/or add more viewpoints, more model-types and artefact-types. However, it is recommended to follow the SCRA, SCRAM and SCIM[5] to be able to use all benefits of the standardization.

6.4 Using the SCRA

The procedure for using the SCRA to take into account the unique needs of a specific smart city (to be carried out by smart cities architecture teams) is the following.

a) Copy the SCRA to the city Solution Architecture (SA).

b) Select the part of this SA to be tailored.

c) Tailor this part of this SA by applying the guidance from the guidance documents.

d) Validate that all parts of this SA are aligned.

e) Proceed to item b) if necessary.

Figure 27 illustrates this process – on the right part of this figure, some SCRA models are reused as-is (green marker) and some SCRA models are tailored (red marker).

[5] SCIM is conceptualized, however its best form is still to be chosen. One of the options is an online IEC Guide.

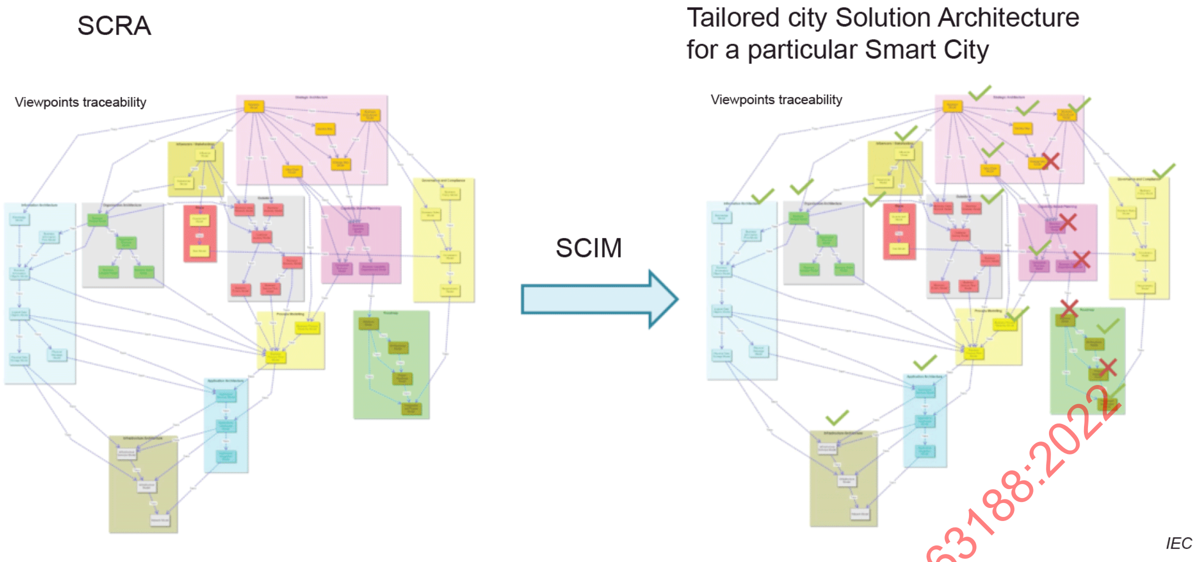

SCRA
Tailored city Solution Architecture for a particular Smart City
Viewpoints traceability
Viewpoints traceability
SCIM
IEC

Figure 27 – Using the SCRA

Potentially, a smart cities architecture team can use the SCRAM, SCRA and SCIM to create their own variant reference architecture, for example, for a particular country.

NOTE 1 The SCRAM and SCRA do not impose or propose a predefined order for defining architecture models because the usual way of creating an architecture description is rather non-linear. Some new information may force an architecture team to start reviewing already prepared architecture models.

NOTE 2 The SCRAM and SCRA do not impose a fixed set of architecture models. Depending on the goals of a smart city, some SCRA architecture models can be ignored, some of them can be changed and some architecture models can be created. However, all such decisions must be recorded in accordance with ISO/IEC/IEEE 42010:2011.

7 SCRAM viewpoints

7.1 General

Each viewpoint is described in accordance with the following template:

- general description (mandatory);
- list of related stakeholders;
- list of related stakeholders' concerns;
- architectural considerations;
- application scenarios;
- list of associated model-types (mandatory to have at least one);
- list of associated KPIs;
- potential improvements;
- list of associated terminological concepts.

NOTE The main stakeholder of the SCRAM viewpoints is IEC SyC Smart Cities.

7.2 Value viewpoint

7.2.1 General description

The value viewpoint describes the problem space in terms of expected value for the stakeholders by providing some ideas about potential overall solutions (called system-solutions to distinguish them from solutions on top of a platform).

7.2.2 List of associated model-types

The value viewpoint comprises the following model-types:

- Problem space description (8.2)
- Problem space terminology (8.3)
- Problem space specifics nomenclature (8.4)
- Problem space classifications nomenclature (8.5)
- Stakeholders nomenclature (8.6)
- Stakeholders' concerns nomenclature (8.7)
- Dependencies between generic stakeholders, stakeholders, stakeholders' concerns and categories of concerns (8.8)
- High-level requirements nomenclature (8.9)
- High-level stories nomenclature (8.10)
- High-level use cases nomenclature (8.11)
- Problem space coverage by the high-level use cases (8.12)
- The mission statement (8.13)
- The vision statement (8.14)
- The strategic goals nomenclature (8.15)

7.3 Big picture viewpoint

7.3.1 General description

The big picture viewpoint outlines the solution space and describes potential system-solutions as a set of high-level elements.

7.3.2 List of associated model-types

The big picture viewpoint comprises the following model-types:

- Solution space boundaries (8.16)
- Solution space terminology (8.17)
- Solution space constraints nomenclature (8.18)
- Solution space classifications nomenclature (8.19)
- Solution space essential desired characteristics nomenclature (8.20)
- Dependency matrix of problem space essential high-level requirements and solution space constraints versus solution space essential desired characteristics (8.21)
- Solution space architecture principles nomenclature (8.22)
- Dependency matrix of solution space essential desired characteristics versus solution space architecture principles (8.23)
- High-level illustrative diagrams (8.24)
- High-level business map (8.25)
- Beneficiaries' journeys nomenclature (8.26)
- High-level reference capability map (8.27)
- High-level process map (8.28)
- High-level architecture (8.29)

7.4 System-solution engineering viewpoint

7.4.1 General description

The system-solution engineering viewpoint describes the system-solution as sets of artefacts such as capabilities, processes, service, functions, data, information, etc. This viewpoint explains how the system-solution delivers value to its beneficiaries as well as how the system-solution actually runs itself.

7.4.2 Architectural considerations

The model-types of the system-solution engineering viewpoint are applied to any potential system-solution as the whole and some of its elements.

7.4.3 List of associated model-types

The system-solution engineering viewpoint comprises the following model-types:

- Capability map (8.30)
- Low-level use cases nomenclature (8.31)
- Function map (8.32)
- Partners nomenclature (8.33)
- Services nomenclature (8.34)
- Process map (8.35)
- Decisions nomenclature (8.36)
- Events nomenclature (8.37)
- Data schemas nomenclature (8.38)
- Information flows nomenclature (8.39)
- Document/content classifications nomenclature (8.40)
- Key performance indicators nomenclature (8.41)
- Reports nomenclature (8.42)
- Platforms nomenclature (8.43)

7.5 Platform engineering viewpoint

7.5.1 General description

The platform engineering viewpoint covers a platform. A potential system-solutions may contain more than one platform.

7.5.2 List of associated model-types

The platform engineering viewpoint comprises the following model-types:

- Platform overview (8.44)
- Platform terminology (8.45)
- Platform components nomenclature (8.46)
- Platform solutions nomenclature (8.47)
- Standards and norms nomenclature (8.48)
- Platform software factory configuration (8.49)
- Platform governance, management and operations manual (8.50)

7.6 Platform component engineering viewpoint

7.6.1 General description

The platform component engineering viewpoint covers a component of a platform.

7.6.2 List of associated model-types

The platform component engineering viewpoint comprises the following model-types:

- Platform component overview (8.51)
- Platform component terminology (8.52)
- Platform component capability map (8.53)
- Platform component function map (8.54)
- Platform component process map (8.55)
- Platform component business specifications (8.56)
- Platform component information specifications (8.57)
- Platform component application specifications (8.58)
- Platform component data specifications (8.59)
- Platform component infrastructure specifications (8.60)
- Platform component security specifications (8.61)
- Platform component services and APIs nomenclature (8.62)
- Platform component software factory configuration (8.63)
- Platform component management and operations manual (8.64)

7.7 Solution engineering viewpoint

7.7.1 General description

The solution engineering viewpoint covers various artefacts of the particular solution.

7.7.2 List of associated model-types

The solution engineering viewpoint comprises the following model-types:

- Solution overview (8.65)
- Solution terminology (8.66)
- Solution capability map (8.67)
- Solution function map (8.68)
- Solution process map (8.69)
- Solution business specifications (8.70)
- Solution information specifications (8.71)
- Solution application specifications (8.72)
- Solution data specifications (8.73)
- Solution infrastructure specifications (8.74)
- Solution security specifications (8.75)
- Solution services and APIs nomenclature (8.76)
- Solution software factory configuration (8.77)
- Solution management and operations manual (8.78)

7.8 Crosscutting aspects engineering viewpoint

7.8.1 General description

The crosscutting aspects engineering viewpoint specifies engineering practices for several essential desired characteristics of the system-solution.

7.8.2 List of associated model-types

The crosscutting aspects viewpoint comprises the following model-types:

- Interoperability aspect (8.79)
- Security aspect (8.80)
- Privacy aspect (8.81)
- Safety aspect (8.82)
- Reliability and resilience management aspects (8.83)
- Low cost of operations and time-to-market aspects (8.84)
- Self-reference aspect (8.85)

7.9 Corporate viewpoint

7.9.1 General description

The corporate viewpoint specifies various governance and management setups of the system-solution.

7.9.2 List of associated model-types

The corporate viewpoint comprises the following model-types:

- Organizational structure (8.86)
- Governance structure (8.87)
- Project management (8.88)
- Corporate manual (8.89)
- Independent evaluation (8.90)
- Ethics, integrity and anti-corruption (8.91)
- Environment and health (8.92)

7.10 Risk management viewpoint

7.10.1 General description

The risk management viewpoint specifies necessary governance practices.

7.10.2 List of associated model-types

The risk management viewpoint comprises the following model-types:

- Risk management aspect (8.93)
- Compliance management aspect (8.94)
- Regulatory management aspects (8.95)
- Crime prevention and detection (8.96)
- Auditing (8.97)
- Independent investigation (8.98)

- Crisis management (8.99)

7.11 Software factory viewpoint

7.11.1 General description

The software factory viewpoint specifies BizDevOps practices. The whole logic of the software factory is to quickly create a simple but complete solution, validate it and gradually enrich it.

7.11.2 List of associated model-types

The software factory viewpoint comprises the following model-types:

- Software factory overview (8.100)
- Prototyping practices (8.101)
- Engineering practices (8.102)
- Assembling practices (8.103)
- Testing practices (8.104)
- Deployment practices (8.105)
- Monitoring practices (8.106)

7.12 Standards viewpoint

7.12.1 General description

The standards viewpoint specifies existing and future standards applicable for the system-solution.

7.12.2 List of associated model-types

The standards viewpoint comprises the following model-types:

- Existing standards nomenclature (see 8.107)
- Potential standards nomenclature (see 8.108)

8 SCRAM model-types

8.1 General

Each model-type is described in accordance with the following template:

- General description (mandatory);
- Architectural considerations;
- Application scenarios;
- Techniques;
- Primary used artefact-types and model-types;
- Recommended activities;
- Guiding materials on how to create a particular model-type from other model-types;
- Primary generated artefact-types;
- Resources;
- Bodies of knowledge pertinent to this model-type;
- Examples;
- Potential improvements;

- List of terminological concepts associated to this model-type.

8.2 Problem space description

8.2.1 General description

The problem space description model outlines some specific phenomena of the problem to be solved.

This model-type is used in the following viewpoints:

- Value viewpoint (7.2)

8.2.2 Techniques

Use the general knowledge of the problem space. Useful techniques can be found in [31].

8.2.3 Primary generated artefact-types

The following artefact-types will be used:

- ProblemSpaceDescription
- ProblemSpaceKeyPhase

8.3 Problem space terminology

8.3.1 General description

Terminology is mandatory to establish efficient communication by defining various concepts pertinent for a particular subject field (e.g. problem space). Ideally, such a terminology should be digital.

This model-type is used in the following viewpoints:

- Value viewpoint (7.2)

8.3.2 Techniques

See the IEC terminological practices.

8.3.3 Smart cities example

The terminology for the smart cities problem domain is under preparation as a separate International Standard.

8.4 Problem space specifics nomenclature

8.4.1 General description

The problem space specifics nomenclature captures various factors which influence the problem space. For example, the political, economic, socio-cultural, technological, environmental and legal factors. They are used later to frame potential solutions. Each of such specifics has a key phase which can be referenced in other model-types.

This model-type is used in the following viewpoints:

- Value viewpoint (7.2)

8.4.2 Techniques

Consider using PEST analysis or similar techniques.

8.4.3 Primary generated artefact-types

The following artefact-types will be used:

- ProblemSpaceSpecificsDescription
- ProblemSpaceSpecificKeyPhase

8.4.4 Smart cities example

See 5.2.

8.5 Problem space classifications nomenclature

8.5.1 General description

Various classifications which may help to structure artefacts are very useful.

This model-type is used in the following viewpoints:

- Value viewpoint (7.2)

8.5.2 Techniques

Analyse the problem space to find existing classifications. It is normal that various classification types may have different structures.

8.5.3 Smart cities example

Typology of smart cities:

- Megapolis
- City
- Town
- Village
- Neighbourhood
- Urban district
- Small island
- Small region
- Valley

8.6 Stakeholders nomenclature

8.6.1 General description

A stakeholder is a person, group of persons or organization having an interest in potential solutions. A list of generic system stakeholders is provided to identify and categorize the stakeholders. Additional classifications of stakeholders may be introduced if necessary.

The stakeholders of the category "Beneficiary" (including primary and secondary) shall be clearly and unambiguously defined in terms of the problem space.

This model-type is used in the following viewpoints:

- Value viewpoint (7.2)

8.6.2 Techniques

Several generic stakeholders (a classification of stakeholders relative to their relationships to a system) are helpful to derive the nomenclature of the stakeholders. For example, the question "who/what is the owner of this system" helps to identify actual stakeholders. The same actual stakeholder may be in several generic stakeholders and vice-versa.

There are four groups of generic stakeholders: system, system-of-systems, environment and ecosystem.

The generic system stakeholders are owner, manager, beneficiary, primary beneficiary, participant, user, architect, developer, tester, installer, liquidator, acquirer, operator and maintainer.

In case of a system-of-systems, it is necessary to consider some additional generic stakeholders, e.g. who coordinates the system-of-systems and who governs it.

Then, generic environment stakeholders shall be considered, for example:

- Professional unions and associations
- Social groups and initiatives
- Regulators
- System partners
- Industry governance body
- Media

Finally, if the system-of-interest belongs to an ecosystem then it is necessary to consider generic ecosystem stakeholders, e.g. who/what governs the ecosystem.

Various classifications can be built for generic stakeholders.

8.6.3 Primary generated artefact-types

The following artefact-types will be used:

- StakeholderTerm
- StakeholderDescription

8.7 Stakeholders' concerns nomenclature

8.7.1 General description

Stakeholders of potential solutions have concerns with respect to the solution considered in relation to its environment. A concern could be held by one or more stakeholders. Concerns arise throughout the life cycle from solution needs and requirements, from architecture choices and from implementation and operating considerations. A concern could be manifest in many forms, such as in relation to one or more stakeholder needs, goals, expectations, responsibilities, requirements, engineering constraints, assumptions, dependencies, quality attributes, architecture decisions, risks or other issues pertaining to potential solutions.

It is mandatory to categorize (or classify) those concerns for further use. Stakeholders' concerns are very important and will be used to determine the high-level requirements.

This model-type is used in the following viewpoints:

- Value viewpoint (7.2)

8.7.2 Techniques

To facilitate the collection of stakeholders' concerns, a set of questions may be prepared with four interrogatives (various colours are used to link interrogatives and placeholders <> within questions): WHO, WHAT (doing something) and WHY (expected outcome) and HOW WELL (measure for success). Such a set shall contain an open (or "runaway") question as shown below. Some of those questions may be used to collect customers' level of interest to define implementation priorities.

Question 1 (positive) What do you want us to improve?

Template: I, as <stakeholder>, want the system to improve <doing something> because <expected outcome> and I will measure the success via <my measure>.

Question 2 (negative) What do you want us to reduce?

Template: I, as <stakeholder>, want the system to reduce <doing something> because <expected outcome > and I will measure the success via <my measure>.

Question 3 (prohibitive) What do you want us to stop?

Template: I, as <stakeholder>, want the system to stop <doing something> because <outcome> and I will measure the success via <my measure>.

Question 4 (runaway) What can we do for you additionally?

Template: I, as <stakeholder>, want the system to start <doing something> because <expected outcome > and I will measure the success via <my measure>.

Stakeholders' concerns may be generated from various groups of stakeholders and various mechanisms, for example:

- direct solicitation;
- crowd wisdom;
- think tanks;
- visionary groups;
- continual improvements;
- staff members' ideas.

8.7.3 Primary generated artefact-types

The following artefact-types will be used:

- ConcernTemplate
- ConcernWhoTerm
- ConcernWhatTerm
- ConcernWhyTerm
- ConcernHowWellTerm

8.7.4 Smart cities example

The following is the potential list of the SCRA stakeholders' concerns:

- Adequate water supply
- Assured electricity supply
- Sanitation, including solid waste management
- Efficient urban mobility and public transport
- Affordable housing, especially for the poor
- Robust IT connectivity and digitization
- Good governance and citizen participation
- Sustainable environment
- Adapted for women, children, people with disabilities and the elderly
- Affordable healthcare for everyone
- Modern education for children and adults
- Attractive for business

8.8 Dependencies between generic stakeholders, stakeholders, stakeholders' concerns and categories of concerns model-type

8.8.1 General description

The various dependencies between architecture generic stakeholders, stakeholders, stakeholders' concerns and categories of concerns can be presented as a line-of-sight diagram. Such diagrams are mandatory to establish the full traceability of decisions and the consistency between various nomenclatures.

For example, a stakeholder without concerns is not really a stakeholder; a concern which is not expressed by any stakeholders is actually not a concern.

This model-type is used in the following viewpoints:

- Value viewpoint (7.2)

8.8.2 Techniques

Line-of-sight diagrams are actually a few dependency matrices. Each such dependency matrix indicates the independent links between two sets of artefacts. Such links are called independent because they don't belong to any of these two sets of artefacts. The presentation of dependency matrix is very straightforward – if there is a dependency between an artefact in a row and an artefact in a column (the former contributes to the latter) then such a dependency is marked in the respective cell.

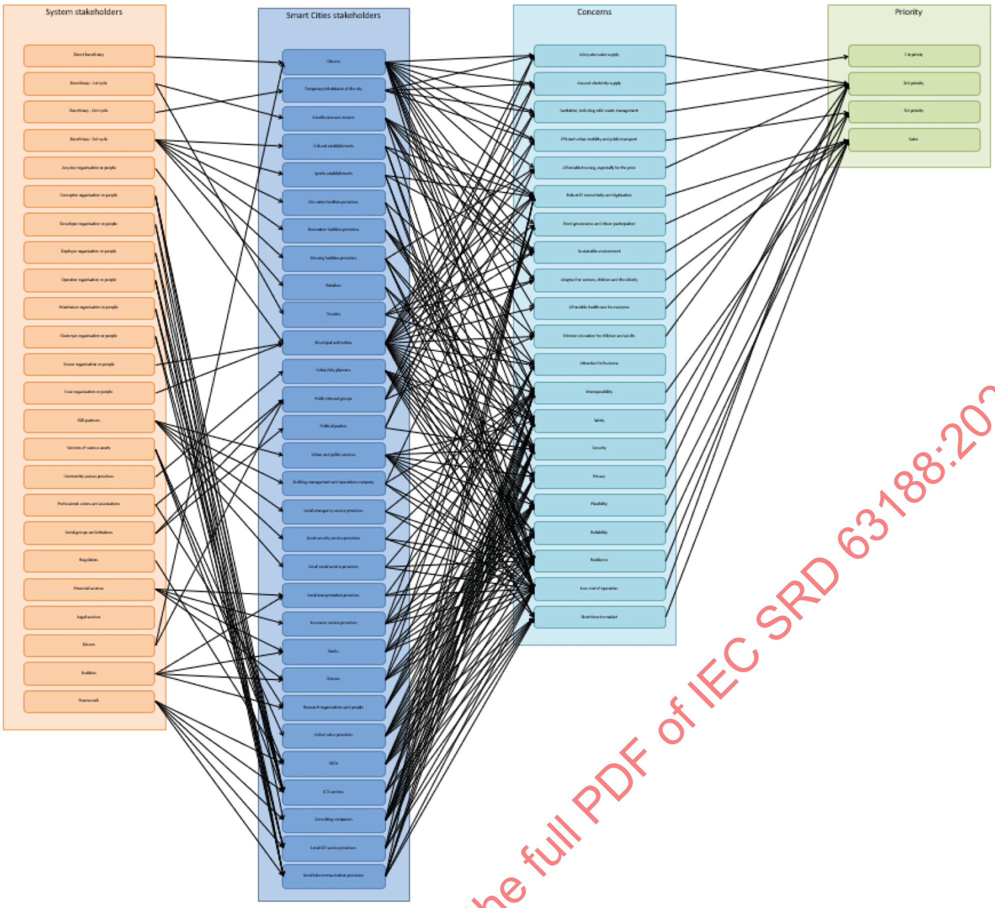
8.8.3 Primary generated artefact-types

The following artefact-types will be used:

- IndependentLink

8.8.4 Smart cities example

Figure 28 shows the dependencies between (from left to right) system stakeholders, stakeholders, stakeholders' concerns and classification of these concerns. One of the recommended classifications is priority which can be used for planning the implementation.


System stakeholders
Direct beneficiary
Beneficiary : 1st cycle
Beneficiary : 2nd cycle
Beneficiary : 3rd cycle
Acquirer organisation or people
Conceptor organisation or people
Developer organisation or people
Deployer organisation or people
Operator organisation or people
Maintainer organisation or people
Destroyer organisation or people
Owner organisation or people
User organisation or people
B2B partners
Vendors of various assets
Commodity services providers
Professional unions and associations
Social groups and initiatives
Regulators
Financial services
Legal services
Drivers
Enablers
Framework
Smart Cities stakeholders
Citizens
Temporary inhabitants of the city
Small businesses owners
Cultural establishments
Sports establishments
Education facilities providers
Recreation facilities providers
Housing facilities providers
Retailers
Tourists
Municipal authorities
Urban/city planners
Public interest groups
Political parties
Urban and public services
Building management and operations company
Local emergency service providers
Local security service providers
Local social service providers
Local transportation providers
Insurance service providers
Banks
Donors
Research organisations and people
Added value providers
SDOs
ICT vendors
Consulting companies
Local ICT service providers
Local telecommunication providers
Concerns
Adequate water supply
Assured electricity supply
Sanitation, including solid waste management
Efficient urban mobility and public transport
Affordable housing, especially for the poor
Robust IT connectivity and digitalisation
Good governance and citizen participation
Sustainable environment
Adapted for women, children and the elderly
Affordable healthcare for everyone
Modern education for children and adults
Attraction for business
Interoperability
Safety
Security
Privacy
Flexibility
Reliability
Resilience
Low cost of operation
Short time to market
Priority
1st priority
2nd priority
3rd priority
Later

IEC

Figure 28 – Dependencies between (from left to right) system stakeholders, stakeholders, stakeholders' concerns and classification of these concerns

8.9 High-level requirements nomenclature

8.9.1 General description

A high-level requirement is an analysed need, expressed by some stakeholders as one of their concerns, related to potential solutions (i.e. some of the stakeholders want those solutions to have some characteristics).

A high-level requirement may be rather complex and crosscutting, e.g. the business may require a city to follow the "Ease of Doing Business" framework [32].

The high-level requirements, the high-level use stories and the high-level use cases are, actually, a step-by-step collection of details about needs of stakeholders.

This model-type is used in the following viewpoints:

- Value viewpoint (7.2)

8.9.2 Techniques

Concerns which are expressed by a particular stakeholder may be used as leads for collecting.

A high-level requirement is captured by three interrogatives (various colours are used to link interrogatives and placeholders <> within questions): WHO, WHAT (doing something) and WHY (expected outcome). A pattern (various colours are used to link interrogatives and placeholders <> within questions) to express high-level requirements is the following:

- As a <person with a particular inability>,
  I want potential solutions that <help me to overcome the limitations of this inability and reduce risks caused by this inability>,
  so that <I can have an active personal, social and professional life>.

Example:

- As a person with a low level of mobility, I want potential solutions that help me to overcome the limitations of this low level of mobility and reduce risks caused by this low level of mobility, thus I can have an active personal, social and professional life.

8.9.3 Recommended activities

- Collect stakeholders' needs based on the stakeholders' concerns.
- Analyse those needs to remove duplications and align terminology.
- Make necessary corrections.
- Validate those corrections with pertinent stakeholders.
- Collect various pieces of information about each high-level requirement, as requested by stakeholders.
- Discuss with or present the results to the stakeholders.

8.9.4 Smart cities example

- Adequate water supply
- Assured electricity supply
- Sanitation, including solid waste management
- Efficient urban mobility and public transport
- Affordable housing, especially for the poor
- Robust IT connectivity and digitalization
- Good governance and citizen participation
- Sustainable environment (also reference to UN Sustainable Development Goal 11)
- Safety and security of citizens, taking into account any specific needs of women, children and the elderly
- Affordable healthcare for everyone
- Modern education for children and adults
- Attractive for business

8.10 High-level stories nomenclature

8.10.1 General description

A high-level story is a high-level requirement in a contextual situation.

The high-level requirements, the high-level user stories and the high-level use cases are, actually a step-by-step collection of details about the needs of stakeholders.

This model-type is used in the following viewpoints:

- Value viewpoint (7.2)

8.10.2 Techniques

A high-level story is captured by five interrogatives (various colours are used to link interrogatives and placeholders <> within questions): WHO, WHAT, WHY, WHERE and WHEN. One high-level requirement may lead to a few high-level stories.

Pattern to express primary beneficiaries' stories is the following:

- As a <person with a particular inability>,
  < in a particular situation>,
  I want potential solutions that will <help me to overcome limitations of this inability and reduce risks caused by this inability>,
  so that <I can have an active personal, social and professional life>.

Examples

- As a person with a low level of mobility, when at home I fall down and lose consciousness, I want potential solutions that will detect and act upon this problem, so that I can be sure that somebody will quickly and professionally help me.
- As a person with a low level of mobility, when I have a problem during my walk in a city park, I want potential solutions that will detect and act upon this problem, thus I can be sure that somebody will quickly and professionally help me.
- As a person with a low level of mobility, when I have a problem during my travel or stay abroad, I want potential solutions that will detect and act upon this problem, thus I can be sure that somebody will quickly and professionally help me.

8.10.3 Recommended activities

- Collect from the stakeholders the high-level stories based on the high-level requirements.
- Analyse those stories to remove duplications and align terminology.
- Make necessary corrections.
- Validate those corrections with pertinent stakeholders.

8.11 High-level use cases nomenclature

8.11.1 General description

A high-level use case is a description of interactions between stakeholders, a potential system-solution or its elements (all as different actors) to achieve the expected outcome as described in a related high-level story. One high-level story may lead to a few high-level use cases.

The high-level requirements, the high-level use stories and the high-level use cases are, actually, a step-by-step collection of details about needs of stakeholders.

This model-type is used in the following viewpoints:

- Value viewpoint (7.2)

8.11.2 Techniques

A high-level use case is captured by six interrogatives (various colours are used to link interrogatives and placeholders <> within questions): WHO, WHAT, WHY, WHERE, WHEN, and HOW, but it does not go into details.

Pattern (various colours are used to link interrogatives and placeholders <> within questions) to express high-level use cases is the following:

- As a <person with a particular inability>,
  <in a particular situation>,
  I want future systems that will <help me to overcome limitations of this inability and reduce risks caused by this inability>,
  by <list of activities>
  so that <I can have an active personal, social and professional life>.

Example:

- As a person with a low level of mobility,
  when at home I fall down and lose consciousness,
  I want future systems that will detect and act upon this problem
  by:
  – all recommended system tools at my home are operational,
  – they detect my problematic situation,
  – they contact my system provider,
  – my provider dispatches the necessary team to my home,
  – this team is provided with enough information about me by my system provider,
  – this team can easily enter my home and find me,
  so that I can be sure that somebody will quickly and professionally help me.

8.11.3 Recommended activities

- Collect from the stakeholders the high-level use cases based on the high-level stories.
- Analyse those use cases to remove duplications and align terminology.
- Make necessary corrections.
- Validate those corrections with pertinent stakeholders.

8.11.4 Primary generated artefact-types

The following artefact-types will be used (HLUC stands for high-level use case):

- HLUC
- HLUCTemplate
- HLUCWhoTerm
- HLUCWhatTerm
- HLUCWhyTerm
- HLUCWhereTerm
- HLUCWhenTerm
- HLUCHowTerm

8.12 Problem space coverage by the high-level use cases

8.12.1 General description

It is strongly recommended to carry out a quality check for the collected high-level use cases. They shall cover the problem space in accordance with the MECE (mutually exclusive and collectively exhaustive) principle.

This model-type is used in the following viewpoints:

- Value viewpoint (7.2)

8.12.2 Techniques

See MECE (mutually exclusive and collectively exhaustive) principle. If the problem space is not covered enough then it is necessary to review the high-level use cases and add some.

8.13 The mission statement

8.13.1 General description

The mission statement describes the problem you are setting out to solve, typically including who you are solving it for.

This model-type is used in the following viewpoints:

- Value viewpoint (7.2)

8.13.2 Techniques

Analyse information collected so far and express in a succinct statement what needs to be done.

8.13.3 Primary generated artefact-types

The following artefact-types will be used:

- MissionDescription
- MissionKeyPhase

8.13.4 Smart cities example

An example of the mission statement for a smart city: "Our mission is to provide outstanding delivery of services to enhance the quality of life for those living in, working in, and visiting our community".

8.14 The vision statement

8.14.1 General description

The vision statement describes an idealized solution that addresses in three to five years perspective the problem you've articulated in the mission.

This model-type is used in the following viewpoints:

- Value viewpoint (7.2)

8.14.2 Techniques

Consider the mission and outline a desired situation to be achieved in three to five years. Strengths, Weaknesses, Opportunities, and Threats (SWOT) analysis can be useful.

8.14.3 Primary generated artefact-types

The following artefact-types will be used:

- VisionDescription
- VisionKeyPhase

8.14.4 Smart cities example

An example of the vision statement for a smart city: "In four years the city will achieve the primary digital elements, knowledge, experience and resources it needs for inclusive and sustainable growth".

8.15 The strategic goals nomenclature

8.15.1 General description

The vision statement may be decomposed into several measurable strategic goals.

This model-type is used in the following viewpoints:

- Value viewpoint (7.2)

8.15.2 Techniques

Typical techniques which can be useful for defining strategic goals are:

- fishbone diagram [33]
- results framework [34]

8.15.3 Smart cities example

See Clause 5.

8.16 Solution space boundaries

8.16.1 General description

Such solution space boundaries separate the solution space from its environment (or context).

The solution space boundaries describe the various known systems, events, actors, etc. around potential system-solutions and other important phenomena, namely:

- "using" systems which employ a potential system-solution;
- "enabling" systems which help a potential system-solution to fulfil its mission;
- "partner" systems which work with a potential system-solution;
- the events outside any potential system-solution that cause a potential system-solution to react;
- the actors outside a potential system-solution that interact with the potential system-solution;
- the information that flows between a potential system-solution and the actors outside it;
- the supply-chains in which a potential system-solution operates.

This model-type is used in the following viewpoints:

- Value viewpoint (7.2)

8.16.2 Techniques

Consider analysing the environment of the system-of-interest and International Requirements Engineering Board (IREB) [35].

8.16.3 Primary generated artefact-types

The following artefact-types will be used:

- SolutionSpaceDescription

8.17 Solution space terminology

8.17.1 General description

Terminology is mandatory to establish efficient communication by defining various concepts pertinent for a particular subject field (e.g. solution space). Ideally, such a terminology should be digital as an ontology.

This model-type is used in the following viewpoints:

- Big picture viewpoint (7.3)

8.17.2 Techniques

See the IEC terminological practices. See ISO/IEC Directives, IEC Supplement:2021, Annex SK.

8.17.3 Smart cities example

The terminology for the smart cities problem domain is under preparation in a separate International Standard.

8.18 Solution space constraints nomenclature

8.18.1 General

Solution space constraints are also known as "guiding principles" or "system requirements" for potential system-solutions. Such constraints are selected to direct potential system-solutions to desired architectural and engineering decisions. The solution space constraints may intersect with some stakeholders' concerns. But the former are more important than the latter because the former are applied to the whole system.

This model-type is used in the following viewpoints:

- Big picture viewpoint (7.3)

8.18.2 Techniques

Constraints may come from different sources – system life cycle, social, environmental, legal, cultural, technological considerations, etc.

8.18.3 Smart cities example

- interoperability by-design and by-default;
- safety by-design and by-default;
- security (including confidentiality, integrity and availability of information) by-design and by-default;
- privacy by-design and by-default;
- reliability and resilience by-design and by-default;
- simplicity;
- low cost of operation;
- short time to market;

- self-referential.

8.18.4 Primary generated artefact-types

The following artefact-types will be used:

- SolutionSpaceConstraintDescription
- SolutionSpaceConstraintKeyPhase

8.19 Solution space classifications nomenclature

8.19.1 General

A solution space may have some classifications which help to organize some artefacts.

This model-type is used in the following viewpoints:

- Big picture viewpoint (7.3)

8.19.2 Techniques

Analyse the solution space to find existing classifications. It is normal that various classification types may have different structures.

8.20 Solution space essential desired characteristics nomenclature

8.20.1 General

Each potential system-solution shall possess some essential desired characteristics to be qualified as a system-solution. This guarantees that each potential system-solution will satisfy solution space constrains and some high-level requirements. A desired characteristic of a system is a characteristic of a system which no system element possesses.

This model-type is used in the following viewpoints:

- Big picture viewpoint (7.3)

8.20.2 Techniques

Defining desired characteristics is a creative activity which uses a lot of knowledge and information including but not limited to stakeholders' concerns, problem space specifics, solution space constraints, etc. Characteristics shouldn't be mixed with functions – the latter are about having abilities to do something, while the former are about some aspects of how it is or will be done.

8.20.3 Primary generated artefact-types

The following artefact-types will be used:

- EssentialDesiredCharacteristicDescription
- EssentialDesiredCharacteristicKeyPhase

8.20.4 Smart cities example

- Be social: Smart cities are built for citizens thus all aspects of social life shall be explicitly considered.
- Be fluid: Cities are considered as systems of flows (see [36]), for example: flows of water, flows of energy, flows of waste, flows of citizens, etc. If nothing flows in a city, then the city is dead. Also, such flows are flows of entities of various types: digital, physical, living, social, political, legal, etc.

- Be multidimensional: Those flows co-exist and interrelate in several dimensions: spatial, temporal, digital, technological, etc. For example, a water sewage canal may be close to a power cable which is a unique supplier for a local hospital. Such multidimensionality shall be explicitly addressed.
- Be scalable: Smart cities are organically grown and shall be extensible and sustainable.
- Be technology neutral: Because of the progress of technology, many various (and unknown right now) intelligent devices (from the IoT) and digital technologies will progressively automate, improve and drastically change various aspects of Smart Cities functioning including planning, execution, monitoring, prediction, optimization of flows.
- Be holistic: Various aspects of the smart cities functioning (e.g. level of security, environmental impact, etc.) shall be integrally (i.e. including all the available data, information and knowledge) anticipated, monitored, analysed, controlled, alerted and acted on.
- Be trustworthy: A high level of trustworthiness (includes security, privacy, safety, reliability, and resilience) is mandatory.

8.21 Dependency matrix of problem space essential high-level requirements and solution space constraints versus solution space essential desired characteristics

8.21.1 General

The dependencies between the problem space essential high-level requirements and solution space constraints and the solution space essential desired characteristics shall be made explicit.

8.21.2 Techniques

Each such dependency matrix indicates the independent links between two sets of artefacts. Such links are called independent because they don't belong to any of these two sets of artefacts. The presentation of dependency matrix is very straightforward – if there is a dependency between an artefact in a row and an artefact in a column (the former contributes to the latter) then such a dependency is marked in the respective cell.

8.21.3 Primary generated artefact-types

The following artefact-types will be used:

- IndependentLink

8.22 Solution space architecture principles nomenclature

8.22.1 General

The architecture principles outline some considerations to be used to achieve the solution space essential desired characteristics.

This model-type is used in the following viewpoints:

- Big picture viewpoint (7.3)

8.22.2 Techniques

Defining of architecture principles is a creative activity which uses a lot of knowledge and information including but not limited to stakeholders' concerns, problem space specifics, solution space constraints, desired characteristics, etc. The following are examples of techniques: TOGAF® [12], Zachman Framework™ [30] and BIZBOK® [37]. [6]

8.22.3 Smart cities example

- Explicit systems architecting and engineering is the best known way to achieve essential characteristics of smart city implementations.
- Smart city is a System of Digital Interrelated Flows which implies total digitalization and intensive use of intelligent devices from the IoT.
- Separation of concerns is mandatory to reduce the complexity of smart city implementations.
- System of Digital Interrelated Flows is an assembly between people, services, applications, devices and organizations.
- System of Digital Interrelated Flows is constructed and operating on the basis of explicit and machine-executable digital contracts.
- Time and place shall be integrated to handle flows properly.
- Terminology is essential because the smart cities system domain covers many, historically disjointed, subject fields.

8.22.4 Primary generated artefact-types

The following artefact-types will be used:

- ArchitecturePrincipleDescription
- ArchitecturePrincipleKeyPhase

8.23 Dependency of between solution space essential desired characteristics versus solution space architecture principles

8.23.1 General

The dependencies between the essential desired characteristics and the solution space architecture principles shall be made explicit.

This model-type is used in the following viewpoints:

- Big picture viewpoint (7.3)

8.23.2 Techniques

Each such dependency matrix indicates the independent links between two sets of artefacts. Such links are called independent because they don't belong to any of these two sets of artefacts. The presentation of dependency matrix is very straightforward – if there is a dependency between an artefact in a row and an artefact in a column (the former contributes to the latter) then such a dependency is marked in the respective cell.

8.23.3 Primary generated artefact-types

The following artefact-types will be used:

- IndependentLink

[6] TOGAF is a trademark of The Open Group. Zachman Framework is a trademark of Zachman International. BIZBOK is a trademark of the Business Architecture Guild. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the products named.

8.24 High-level illustrative diagrams

8.24.1 General

Illustrative diagrams are informal presentations of potential system-solutions.

This model-type is used in the following viewpoints:

- Big picture viewpoint (7.3)

8.24.2 Techniques

There are three essential parts in such illustrations: the important entities in the environment of potential system-solutions, the essential elements of potential system-solutions, and relationships between them.

8.24.3 Smart cities example

See Figure 29.

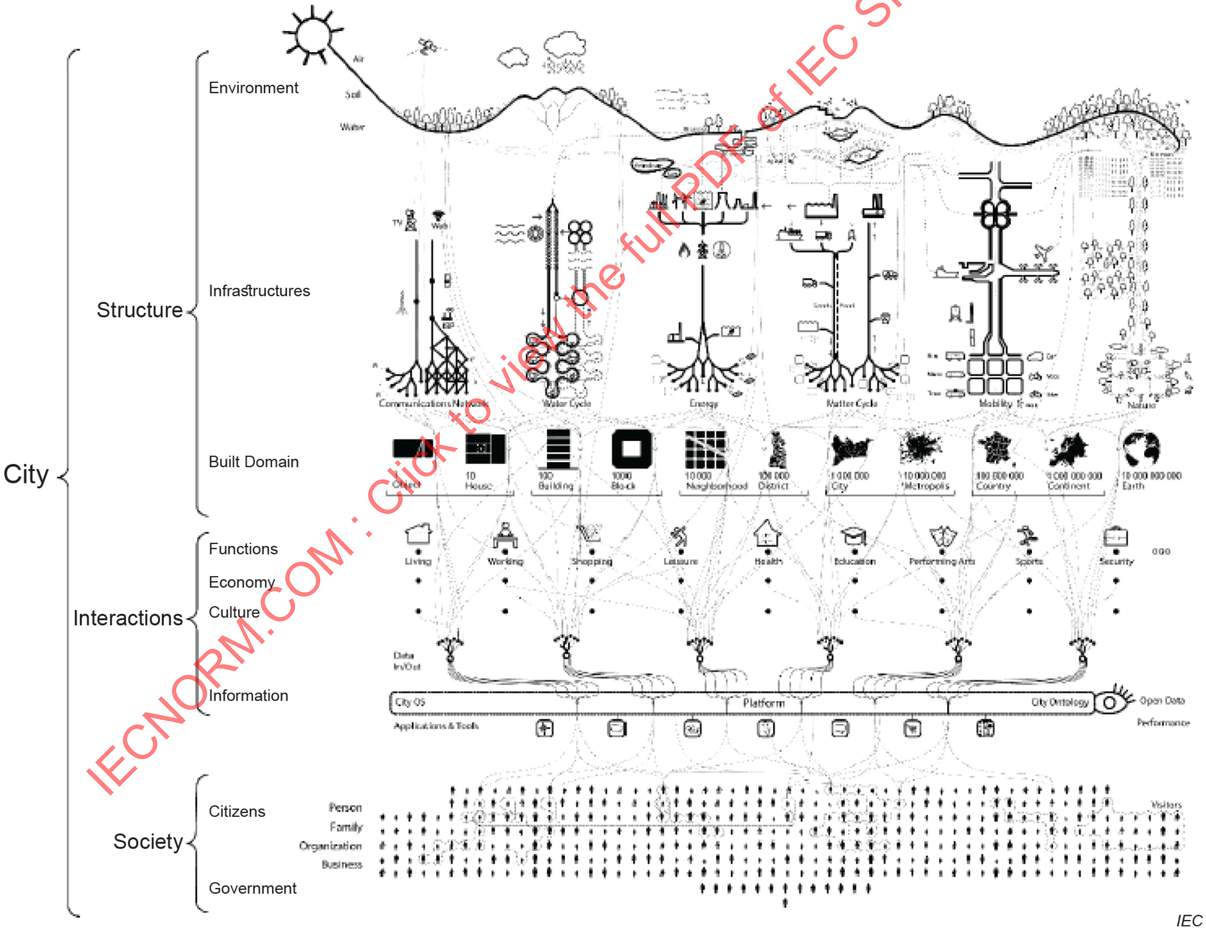


Figure 29 – Descriptive framework from ISO 37105:2019

8.25 High-level business map

8.25.1 General

The high-level business model is a sort of plan for the successful operation of a system-solution to fulfil its vision, identifying sources of revenue, the intended customer base, products, and details of financing.

This model-type is used in the following viewpoints:

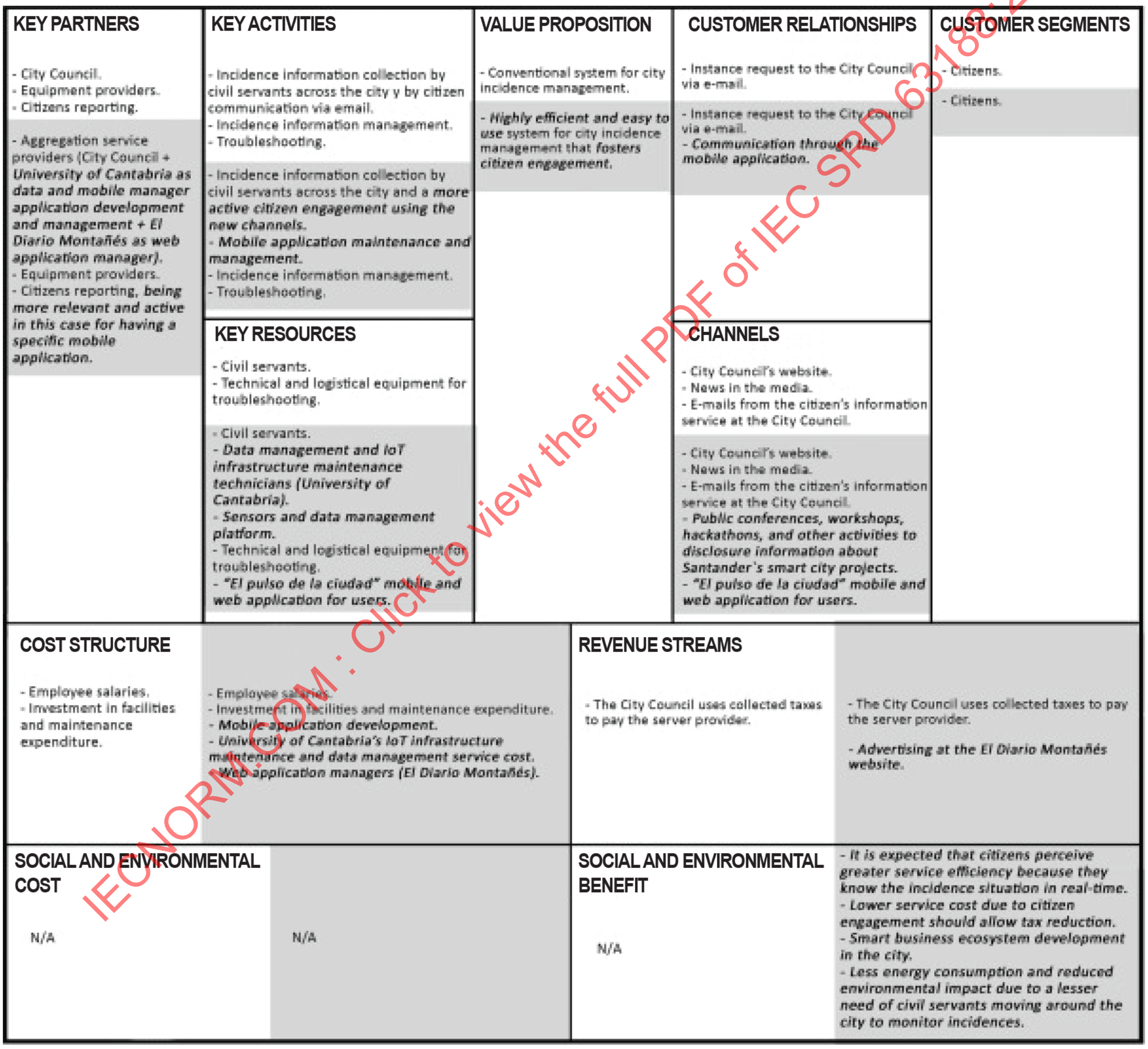
- Big picture viewpoint (7.3)

8.25.2 Techniques

The method "Business Canvas" may be used to express a business map. Because, this technique is a collection of existing artefact-types (from different model-types), it may be considered as a thinking tool to be enriched and validated by further development of those artefact-types.

8.25.3 Smart cities example

Figure 30 shows a Business Canvas for smart city (it is copied from [38]).



KEY PARTNERS

- City Council.
- Equipment providers.
- Citizens reporting.

- Aggregation service providers (City Council + *University of Cantabria as data and mobile manager application development and management + El Diario Montañés as web application manager).*
- Equipment providers.
- Citizens reporting, *being more relevant and active in this case for having a specific mobile application.*

KEY ACTIVITIES

- Incidence information collection by civil servants across the city y by citizen communication via email.
- Incidence information management.
- Troubleshooting.

- Incidence information collection by civil servants across the city and a *more active citizen engagement using the new channels.*
- *Mobile application maintenance and management.*
- Incidence information management.
- Troubleshooting.

KEY RESOURCES

- Civil servants.
- Technical and logistical equipment for troubleshooting.

- Civil servants.
- *Data management and IoT infrastructure maintenance technicians (University of Cantabria).*
- *Sensors and data management platform.*
- Technical and logistical equipment for troubleshooting.
- *"El pulso de la ciudad" mobile and web application for users.*

VALUE PROPOSITION

- Conventional system for city incidence management.

- *Highly efficient and easy to use* system for city incidence management that *fosters citizen engagement.*

CUSTOMER RELATIONSHIPS

- Instance request to the City Council via e-mail.

- Instance request to the City Council via e-mail.
- *Communication through the mobile application.*

CHANNELS

- City Council's website.
- News in the media.
- E-mails from the citizen's information service at the City Council.

- City Council's website.
- News in the media.
- E-mails from the citizen's information service at the City Council.
- *Public conferences, workshops, hackathons, and other activities to disclosure information about Santander's smart city projects.*
- *"El pulso de la ciudad" mobile and web application for users.*

CUSTOMER SEGMENTS

- Citizens.

- Citizens.

COST STRUCTURE

- Employee salaries.
- Investment in facilities and maintenance expenditure.

- Employee salaries.
- Investment in facilities and maintenance expenditure.
- *Mobile application development.*
- *University of Cantabria's IoT infrastructure maintenance and data management service cost.*
- *Web application managers (El Diario Montañés).*

REVENUE STREAMS

- The City Council uses collected taxes to pay the server provider.

- The City Council uses collected taxes to pay the server provider.

- *Advertising at the El Diario Montañés website.*

SOCIAL AND ENVIRONMENTAL COST

N/A

N/A

SOCIAL AND ENVIRONMENTAL BENEFIT

N/A

- *It is expected that citizens perceive greater service efficiency because they know the incidence situation in real-time.*
- *Lower service cost due to citizen engagement should allow tax reduction.*
- *Smart business ecosystem development in the city.*
- *Less energy consumption and reduced environmental impact due to a lesser need of civil servants moving around the city to monitor incidences.*

IEC

Figure 30 – Business Canvas model for smart city

8.26 Beneficiaries' journeys nomenclature

8.26.1 General description

An integrated presentation of the behaviour of some beneficiaries, also known as "customer journeys", may be useful. Such journeys are, actually, assemblies of several high-level use cases related to the same stakeholder.

This model-type is used in the following viewpoints:

- Value viewpoint (7.2)

8.26.2 Techniques

Any smart city is built around the citizens' experience. By understanding the tasks the beneficiaries perform a smart city should be able to align its capabilities to meet the beneficiaries' needs. This may include:

- The beneficiary task
- Desired outcomes when performing the task
- Constraints which need to be considered
- Beneficiary experience when performing the task
- Product innovations required to support the task
- Products and services these innovations relate to

8.26.3 Primary generated artefact-types

The following artefact-types will be used:

- JourneyTemplate

8.27 High-level reference capability map

8.27.1 General

This is a map of reference capabilities of potential system-solutions. See 5.19.3.

This model-type is used in the following viewpoints:

- Big picture viewpoint (7.3)

8.27.2 Techniques

See 5.19.3.

8.27.3 Primary generated artefact-types

The following artefact-types will be used:

- ReferenceCapability

8.27.4 Smart cities example

See Figure 31.

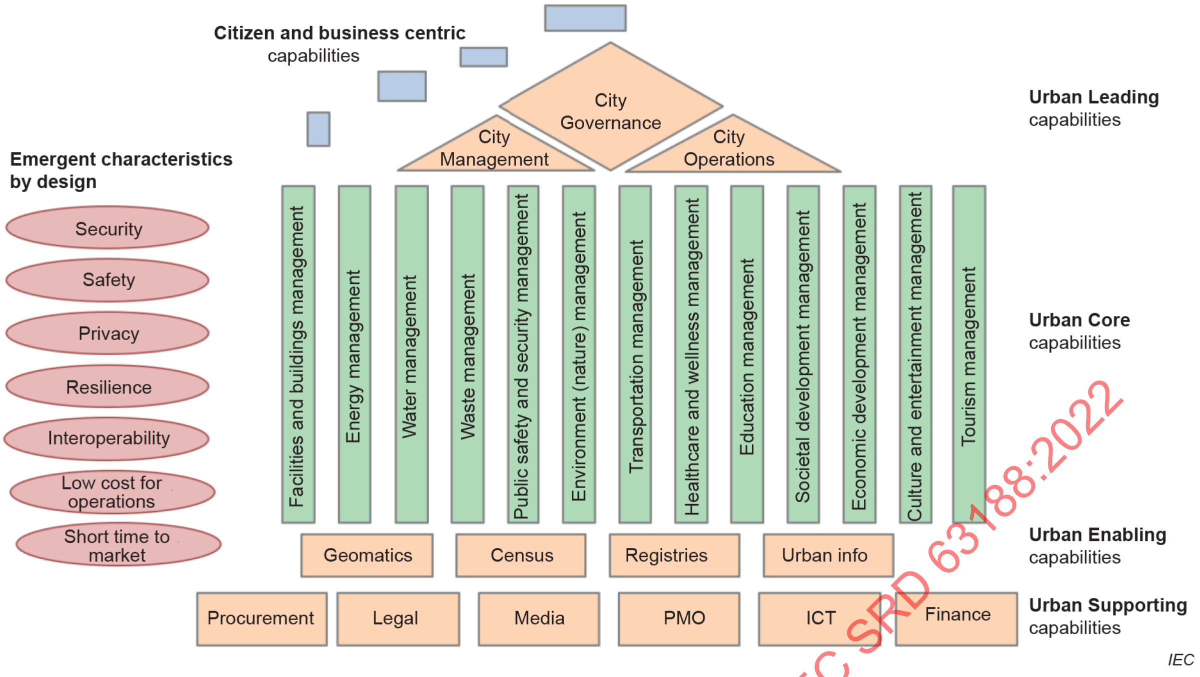

Citizen and business centric capabilities
Emergent characteristics by design
Security
Safety
Privacy
Resilience
Interoperability
Low cost for operations
Short time to market
City Governance
City Management
City Operations
Urban Leading capabilities
Facilities and buildings management
Energy management
Water management
Waste management
Public safety and security management
Environment (nature) management
Transportation management
Healthcare and wellness management
Education management
Societal development management
Economic development management
Culture and entertainment management
Tourism management
Urban Core capabilities
Geomatics
Census
Registries
Urban info
Urban Enabling capabilities
Procurement
Legal
Media
PMO
ICT
Finance
Urban Supporting capabilities
IEC

Figure 31 – Potential SCRA level 1 reference capability map

8.28 High-level process map

8.28.1 General

This is a map of top-level processes of potential system-solutions. See 5.19.3.

This model-type is used in the following viewpoints:

- Big picture viewpoint (7.3)

8.28.2 Techniques

The following techniques may be applied: ISO/IEC 19510:2013 [23], value stream mapping and the APQC Process Classification Framework (PCF)®[7] [39]. Some process modelling conventions will be provided in the SCIM.

8.28.3 Primary generated artefact-types

The following artefact-types will be used:

- ProcessDiagram

8.29 High-level architecture

8.29.1 General

The high-level architecture formally defines a top-level structure of the potential system-solutions (top-level elements and relationships between them) and some relationships of those top-level elements with its environments. Each of such elements contain one or more reference capabilities.

This model-type is used in the following viewpoints:

[7] Process Classification Framework (PCF) is a trademark of APQC. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named.

- Big picture viewpoint (7.3)

8.29.2 Techniques

Providing the high-level architecture is a creative work which requires a lot of analysis of the collected information and a synthesis of the high-level architecture. The following are examples of techniques: TOGAF [12] and BIZBOK [36].

8.30 Capability map

8.30.1 General description

This is a map of capabilities of the system-solution. See 5.19.3. The capability map acts as an "accumulator" of all identified capabilities.

This model-type is used in the following viewpoints:

- System-solution engineering viewpoint (7.4)

8.30.2 Techniques

See 5.19.3.

8.30.3 Primary generated artefact-types

The following artefact-types will be used:

- Capability

8.31 Low-level use cases nomenclature

8.31.1 General description

A low-level use case is a structured description of a particular use of one or many capabilities of the system-solution by one or many stakeholders.

This model-type is used in the following viewpoints:

- System-solution engineering viewpoint (7.4)

8.31.2 Techniques

A use case technique (which is actually a linking of various model-types and artefact-types) should be based on IEC 62559-2 [40]. The following are also examples of techniques: BABOK® [41] and UML® [42].[8]

8.31.3 Primary generated artefact-types

The following artefact-types will be used:

- LLUC

[8] BABOK is a trademark of the International Institute of Business Analysis. UML is a trademark of the Object Management Group. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the products named.

8.32 Function map

8.32.1 General description

This is a map of functions of the system-solution. See 5.19.3. The function map acts as an "accumulator" of all identified functions.

This model-type is used in the following viewpoints:

- System-solution engineering viewpoint (7.4)

8.32.2 Techniques

See 5.19.3

8.32.3 Primary generated artefact-types

The following artefact-types will be used:

- Function

8.33 Partners nomenclature

8.33.1 General description

Because some of the system-solution capabilities may be sourced by a solution partner (an organization from the environment of the particular smart city), a nomenclature of such partners and relationships with them is necessary.

This model-type is used in the following viewpoints:

- System-solution engineering viewpoint (7.4)

8.33.2 Techniques

Describe a company and all the contracts with it.

8.33.3 Primary generated artefact-types

The following artefact-types will be used:

- SolutionPartnerOrganization

8.34 Services nomenclature

8.34.1 General description

This is a nomenclature of all services of the system-solution. See 5.19.4. The services nomenclature acts as an "accumulator" of all identified services.

This model-type is used in the following viewpoints:

- System-solution engineering viewpoint (7.4)

8.34.2 Techniques

There are many techniques to identify services, e.g. Service-Oriented Architecture (SOA) [43].

8.34.3 Primary generated artefact-types

The following artefact-types will be used:

- Service

8.34.4 Smart cities example

Figure 32 is from the article "smart city Concept, Applications and Services" [44].

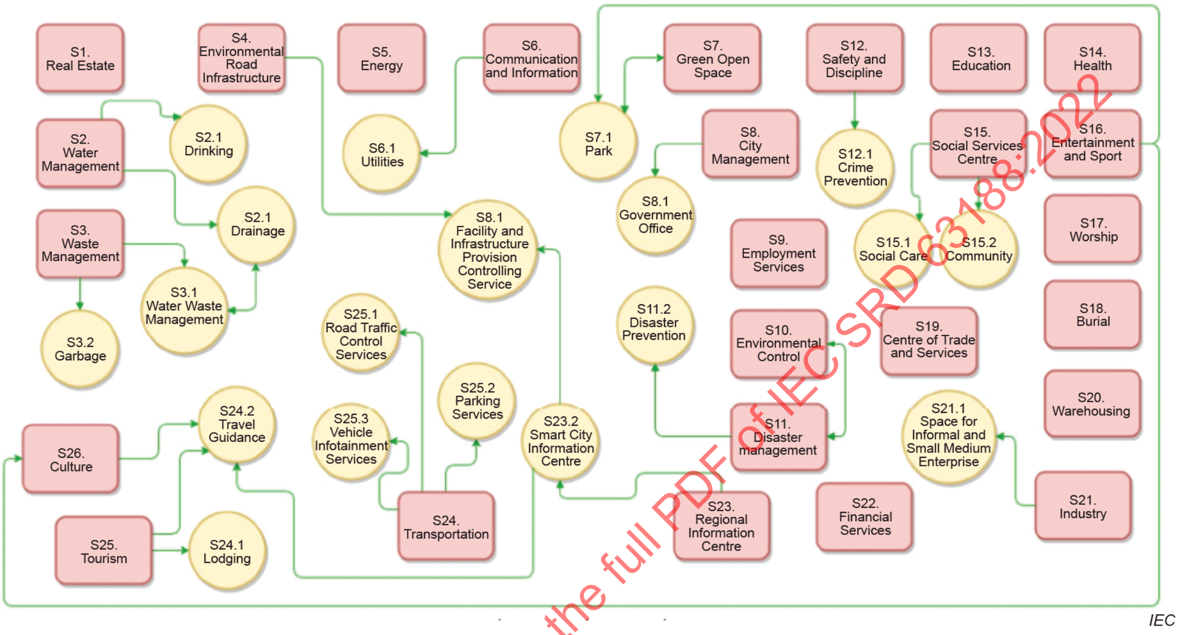

S1. Real Estate
S4. Environmental Road Infrastructure
S5. Energy
S6. Communication and Information
S7. Green Open Space
S12. Safety and Discipline
S13. Education
S14. Health
S2. Water Management
S2.1 Drinking
S6.1 Utilities
S7.1 Park
S8. City Management
S12.1 Crime Prevention
S15. Social Services Centre
S16. Entertainment and Sport
S3. Waste Management
S2.1 Drainage
S8.1 Facility and Infrastructure Provision Controlling Service
S8.1 Government Office
S9. Employment Services
S15.1 Social Care
S15.2 Community
S17. Worship
S3.2 Garbage
S3.1 Water Waste Management
S25.1 Road Traffic Control Services
S11.2 Disaster Prevention
S10. Environmental Control
S19. Centre of Trade and Services
S18. Burial
S24.2 Travel Guidance
S25.3 Vehicle Infotainment Services
S25.2 Parking Services
S23.2 Smart City Information Centre
S11. Disaster management
S21.1 Space for Informal and Small Medium Enterprise
S20. Warehousing
S26. Culture
S25. Tourism
S24.1 Lodging
S24. Transportation
S23. Regional Information Centre
S22. Financial Services
S21. Industry
IEC

Figure 32 – Example of city services

8.35 Process map

8.35.1 General description

This is a map of processes of the system-solution. See 5.19.3. The process map acts as an "accumulator" of all identified processes.

This model-type is used in the following viewpoints:

- System-solution engineering viewpoint (7.4)

8.35.2 Techniques

The following techniques may be applied: ISO/IEC 19510:2013 [23], value stream mapping and the APQC Process Classification Framework (PCF) [39]. Some process modelling conventions will be provided in the SCIM.

8.35.3 Primary generated artefact-types

The following artefact-types will be used:

- ProcessDiagram

8.36 Decisions nomenclature

8.36.1 General description

Decisions are formally defined rules that govern and manage the system-solution. The decision nomenclature acts as an "accumulator" of all identified decision types.

This model-type is used in the following viewpoints:

- System-solution engineering viewpoint (7.4)

8.36.2 Techniques

Decision Model and Notation (DMN™)[9] [45] is an available body of knowledge.

8.36.3 Primary generated artefact-types

The following artefact-types will be used:

- Decision

8.37 Events nomenclature

8.37.1 General description

Event is an incident of importance to the system-solution. The events nomenclature acts as an "accumulator" of all identified event types.

This model-type is used in the following viewpoints:

- System-solution engineering viewpoint (7.4)

8.37.2 Techniques

Event Driven Architecture (EDA) and Event Processing Network (EPN) are available bodies of knowledge.

8.37.3 Primary generated artefact-types

The following artefact-types will be used:

- Event

8.38 Data schemas nomenclature

8.38.1 General description

Data schema is a structure of data objects. The data schemas nomenclature acts as an "accumulator" of all identified data schemas.

This model-type is used in the following viewpoints:

- System-solution engineering viewpoint (7.4)

8.38.2 Techniques

The Data Management Association International (DAMA) [46] is an available body of knowledge.

[9] DMN is an abbreviation and a trademark of the Object Management Group. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named.

8.38.3 Primary generated artefact-types

The following artefact-types will be used:

- DataSchema

8.39 Information flows nomenclature

8.39.1 General description

Information flow is an established channel for information exchange between system elements and with external entities. The information flow nomenclature acts as an "accumulator" of all identified information flows.

This model-type is used in the following viewpoints:

- System-solution engineering viewpoint (7.4)

8.39.2 Techniques

The Open Group is an available body of knowledge.

8.39.3 Primary generated artefact-types

The following artefact-types will be used:

- InformationFlow

8.40 Document/content classifications nomenclature

8.40.1 General description

Classification is a typology of some objects to be recognized, differentiated, and understood. The document/content classification nomenclature acts as an "accumulator" of all identified document/content classifications.

This model-type is used in the following viewpoints:

- System-solution engineering viewpoint (7.4)

8.40.2 Techniques

Association for Information and Image Management (AIIM) [47] is an available body of knowledge.

8.40.3 Primary generated artefact-types

The following artefact-types will be used:

- DocumentClassification
- ContentClassifiation

8.41 Key performance indicators nomenclature

8.41.1 General description

Key Performance Indicator (KPI) is a set of values against which performance is measured. The key performance indicators nomenclature acts as an "accumulator" of all identified key performance indicators.

This model-type is used in the following viewpoints:

- System-solution engineering viewpoint (7.4)

8.41.2 Techniques

The Software Performance Engineering (SPE) [48] is an available body of knowledge.

8.41.3 Primary generated artefact-types

The following artefact-types will be used:

- KeyPerformanceIndicator

8.42 Reports nomenclature

8.42.1 General description

Report is an extracted and formatted information. The report nomenclature acts as an "accumulator" of all identified reports.

This model-type is used in the following viewpoints:

- System-solution engineering viewpoint (7.4)

8.42.2 Primary generated artefact-types

The following artefact-types will be used:

- Report

8.43 Platforms nomenclature

8.43.1 General description

The platforms nomenclature lists all the platforms which are used in the system-solution.

This model-type is used in the following viewpoints:

- Platform engineering viewpoint (7.4)

8.43.2 Primary generated artefact-types

The following artefact-types will be used:

- Platform,
- PlatformComponent.

8.44 Platform overview

8.44.1 General description

The platform overview provides general information about the platform including a list of its components and a list of solutions which are built on top of the platform.

This model-type is used in the following viewpoints:

- Platform engineering viewpoint (7.5)

8.44.2 Techniques

"Software Architecture in Practice, Second Edition" [49] is an available body of knowledge.

8.44.3 Primary generated artefact-types

The following artefact-types will be used:

- Platform,
- PlatformComponent.

8.45 Platform terminology

8.45.1 General description

Terminology is mandatory to establish efficient communication by defining various concepts pertinent for a particular subject field (e.g. platform). Ideally, such a terminology should be digital as an ontology.

This model-type is used in the following viewpoints:

- Platform engineering viewpoint (7.5)

8.45.2 Techniques

See the IEC terminological practices, see ISO/IEC Directives, IEC Supplement:2021, Annex SK.

8.46 Platform components nomenclature

8.46.1 General description

The platform components nomenclature lists all the platform components which are used in a particular platform.

This model-type is used in the following viewpoints:

- Platform engineering viewpoint (7.5)

8.46.2 Primary generated artefact-types

The following artefact-types will be used:

- PlatformComponent

8.47 Platform solutions nomenclature

8.47.1 General description

The platform solutions nomenclature lists all the solutions which are built on top of the platform.

This model-type is used in the following viewpoints:

- Platform engineering viewpoint (7.5)

8.47.2 Primary generated artefact-types

The following artefact-types will be used:

- Solution

8.48 Standards and norms nomenclature

8.48.1 General description

Standards and norms are formal rules to be applied within the life cycle of the system-solution.

This model-type is used in the following viewpoints:

- System-solution engineering viewpoint (7.5)

8.48.2 Primary generated artefact-types

The following artefact-types will be used:

- InternalStandard
- InternalNorm

8.49 Platform software factory configuration

8.49.1 General description

The platform software factory configuration describes all the details for platform software development, testing, deployment, execution and decommissioning.

This model-type is used in the following viewpoints:

- Platform engineering viewpoint (7.5)

8.49.2 Techniques

The platform software factory configuration is based on the software factory viewpoint (7.11).

8.49.3 Primary generated artefact-types

The following artefact-types will be used:

- SoftwareFactoryConfiguration

8.50 Platform governance, management and operations manual

8.50.1 General description

The platform governance, management and operations manual provides necessary information for efficient use and evolution of the platform.

This model-type is used in the following viewpoints:

- Platform engineering viewpoint (7.5)

8.50.2 Techniques

The best organization of such a manual is a wiki-like structure and functionality.

8.50.3 Primary generated artefact-types

The following artefact-types will be used:

- WikiManual

8.51 Platform component overview

8.51.1 General description

The platform component overview provides general information about the platform component.

This model-type is used in the following viewpoints:

- Platform component engineering viewpoint (7.6)

8.51.2 Techniques

"Software Architecture in Practice, Second Edition" [49] is an available body of knowledge.

8.51.3 Primary generated artefact-types

The following artefact-types will be used:

- PlatformComponent

8.52 Platform component terminology

8.52.1 General description

Terminology is mandatory to establish efficient communication by defining various concepts pertinent for a particular subject field (e.g. platform component). Ideally, such a terminology should be digital as an ontology.

This model-type is used in the following viewpoints:

- Platform component engineering viewpoint (7.6)

8.52.2 Techniques

See the IEC terminological practices.

8.53 Platform component capability map

8.53.1 General description

This is a map of capabilities of the platform component. See 5.19.3.

This model-type is used in the following viewpoints:

- Platform component engineering viewpoint (7.6)

8.53.2 Techniques

See 5.19.3.

8.53.3 Primary generated artefact-types

The following artefact-types will be used:

- Capability

8.54 Platform component function map

8.54.1 General description

This is a map of functions of the platform component. See 5.19.3.

This model-type is used in the following viewpoints:

- Platform component engineering viewpoint (7.6)

8.54.2 Techniques

See 5.19.3.

8.54.3 Primary generated artefact-types

The following artefact-types will be used:

- Function

8.55 Platform component process map

8.55.1 General description

This is a map of processes of the platform component. See 5.19.3.

This model-type is used in the following viewpoints:

- Platform component engineering viewpoint (7.6)

8.55.2 Techniques

ISO/IEC 19510:2013 [23], value stream mapping and the APQC Process Classification Framework (PCF) [39] are available bodies of knowledge. Some process modelling conventions will be provided in the SCIM.

8.55.3 Primary generated artefact-types

The following artefact-types will be used:

- ProcessDiagram

8.56 Platform component business specifications

8.56.1 General description

The platform component business specifications cover the following parts of the platform component:

- platform component stakeholders and roles,
- platform component use cases,
- platform component activities,
- platform component decisions,
- platform component UI,
- platform component key performance indicators
- platform component events
- platform component reports

This model-type is used in the following viewpoints:

- Platform component engineering viewpoint (7.6)

8.56.2 Techniques

The Zachman Framework [30] is an available body of knowledge.

8.56.3 Primary generated artefact-types

The following artefact-types will be used:

- LLUC,
- Decision
- UI,

- KPI,
- Event,
- Report.

8.57 Platform component information specifications

8.57.1 General description

The platform component information specifications cover the following user-accessible information of the platform component:

- geomatics-specific information,
- content-specific information,
- analytics-specific information,
- reporting-specific information,
- social-specific information,
- other information.

This model-type is used in the following viewpoints:

- Platform component engineering viewpoint (7.6)

8.57.2 Techniques

Data Management Association International (DAMA) [45], and Association for Information and Image Management (AIIM) [47] are available bodies of knowledge.

8.57.3 Primary generated artefact-types

The following artefact-types will be used:

- DataSchema
- InformationFlow

8.58 Platform component application specifications

8.58.1 General description

The platform component application specifications cover the following software-specific parts of the platform component:

- internal software components,
- external software components,
- software protocols between software components.

This model-type is used in the following viewpoints:

- Platform component engineering viewpoint (7.6)

8.58.2 Techniques

Software engineering books

8.58.3 Primary generated artefact-types

The following artefact-types will be used:

- SoftwareComponent,
- SoftwareProtocol.

8.59 Platform component data specifications

8.59.1 General description

The platform component data specifications cover the following data-specific parts of the platform component:

- data storage,
- data transit, and
- data usage.

This model-type is used in the following viewpoints:

- Platform component engineering viewpoint (7.6)

8.59.2 Techniques

Data Management Association International (DAMA) [46] is an available body of knowledge.

8.59.3 Primary generated artefact-types

The following artefact-types will be used:

- DataSchema
- InformationFlow

8.60 Platform component infrastructure specifications

8.60.1 General description

The platform component data specifications cover the following infrastructure-specific parts of the platform component:

- gateways,
- servers,
- database,
- backup policies,
- environments,
- etc.

This model-type is used in the following viewpoints:

- Platform component engineering viewpoint (7.6)

8.60.2 Techniques

ITIL [50] and ITSM [51] are available bodies of knowledge.

8.60.3 Primary generated artefact-types

The following artefact-types will be used:

- TechnicalDiagram

8.61 Platform component security specifications

8.61.1 General description

The platform component security specifications cover the following crosscutting aspects of the platform component:

- security,
- privacy,
- safety, and
- reliability and resilience.

This model-type is used in the following viewpoints:

- Platform component engineering viewpoint (7.6)

8.61.2 Techniques

See 5.19.

8.61.3 Primary generated artefact-types

The following artefact-types will be used:

- GenericSpecification

8.62 Platform component services and APIs nomenclature

8.62.1 General description

This is a nomenclature of services and APIs of the platform component. See 5.19.4.

This model-type is used in the following viewpoints:

- Platform component engineering viewpoint (7.6)

8.62.2 Techniques

See 5.19.4.

8.62.3 Primary generated artefact-types

The following artefact-types will be used:

- Service
- API

8.63 Platform component software factory configuration

8.63.1 General description

The platform software factory configuration describes all the details of an element (e.g. platform component) related to its software development, testing, deployment and execution.

This model-type is used in the following viewpoints:

- Platform component engineering viewpoint (7.6)

8.63.2 Techniques

The platform component software factory configuration is based on the software factory viewpoint (7.11).

8.63.3 Primary generated artefact-types

The following artefact-types will be used:

- SoftwareConfiguration

8.64 Platform component management and operations manual

8.64.1 General description

The platform component management and operations manual provides necessary information for the efficient evolution of the platform component.

This model-type is used in the following viewpoints:

- Platform component engineering viewpoint (7.6)

8.64.2 Techniques

The best organization of such a manual is a wiki-like structure and functionality.

8.64.3 Primary generated artefact-types

The following artefact-types will be used:

- WikiManual

8.65 Solution overview

8.65.1 General description

The solution overview provides general information about the solution.

This model-type is used in the following viewpoints:

- Solution engineering viewpoint (7.7)

8.65.2 Techniques

"Software Architecture in Practice, Second Edition" [49] is an available body of knowledge.

8.65.3 Primary generated artefact-types

The following artefact-types will be used:

- PlatformComponent

8.66 Solution terminology

8.66.1 General description

Terminology is mandatory to establish efficient communication by defining various concepts pertinent for a particular subject field (e.g. solution). Ideally, such a terminology should be digital.

This model-type is used in the following viewpoints:

- Solution engineering viewpoint (7.7)

8.66.2 Techniques

See the IEC terminological practices.

8.67 Solution capability map

8.67.1 General description

This is a map of capabilities of the solution. See 5.19.3.

This model-type is used in the following viewpoints:

- Solution engineering viewpoint (7.7)

8.67.2 Techniques

See 5.19.3.

8.67.3 Primary generated artefact-types

The following artefact-types will be used:

- Capability

8.68 Solution function map

8.68.1 General description

This is a map of functions of the solution. See 5.19.3.

This model-type is used in the following viewpoints:

- Solution engineering viewpoint (7.7)

8.68.2 Techniques

See 5.19.3.

8.68.3 Primary generated artefact-types

The following artefact-types will be used:

- Function

8.69 Solution process map

8.69.1 General description

This is a map of processes of the solution. See 5.19.3.

This model-type is used in the following viewpoints:

- Solution engineering viewpoint (7.7)

8.69.2 Techniques

The following techniques can be applied: ISO/IEC 19510:2013 [23], value stream mapping and the APQC Process Classification Framework (PCF) [39]. Some process modelling conventions will be provided in the SCIM.

8.69.3 Primary generated artefact-types

The following artefact-types will be used:

- ProcessDiagram

8.70 Solution business specifications

8.70.1 General description

The solution business specifications cover the following parts of the solution:

- solution stakeholders and roles,
- solution use cases,
- solution activities,
- solution decisions,
- solution UI,
- solution key performance indicators
- solution events
- solution reports

This model-type is used in the following viewpoints:

- Solution engineering viewpoint (7.7)

8.70.2 Techniques

The Zachman Framework [30] is an available body of knowledge.

8.70.3 Primary generated artefact-types

The following artefact-types will be used:

- LLUC,
- Decision,
- UI,
- KPI,
- Event,
- Report.

8.71 Solution information specifications

8.71.1 General description

The solution information specifications cover the following user-accessible information of the solution:

- geomatics-specific information,
- content-specific information,
- analytics-specific information,
- reporting-specific information,
- social-specific information,
- other information.

This model-type is used in the following viewpoints:

- Solution engineering viewpoint (7.7)

8.71.2 Techniques

Data Management Association International (DAMA) [46], and the Association for Information and Image Management (AIIM) [47] are available bodies of knowledge.

8.72 Solution application specifications

8.72.1 General description

The solution application specifications cover the following software-specific parts of the solution:

- internal software components,
- external software components,
- software protocols between software components.

This model-type is used in the following viewpoints:

- Solution engineering viewpoint (7.7)

8.72.2 Techniques

Software engineering books

8.72.3 Primary generated artefact-types

The following artefact-types will be used:

- SoftwareComponent,
- SoftwareProtocol.

8.73 Solution data specifications

8.73.1 General description

The solution data specifications cover the following data-specific parts of the solution:

- data storage,
- data transit, and
- data usage.

This model-type is used in the following viewpoints:

- Solution engineering viewpoint (7.7)

8.73.2 Techniques

Data Management Association International (DAMA) [46] is an available body of knowledge.

8.74 Solution infrastructure specifications

8.74.1 General description

The solution data specifications cover the following infrastructure-specific parts of the solution:

- gateways,
- servers,
- database,
- backup policies,
- environments,
- etc.

This model-type is used in the following viewpoints:

- Solution engineering viewpoint (7.7)

8.74.2 Techniques

ITIL [50] and ITSM [51] are available bodies of knowledge.

8.74.3 Primary generated artefact-types

The following artefact-types will be used:

- TechnicalDiagram

8.75 Solution security specifications

8.75.1 General description

The solution security specifications cover the following crosscutting aspects of the solution:

- security,
- privacy,
- safety, and
- reliability and resilience.

This model-type is used in the following viewpoints:

- Solution engineering viewpoint (7.7)

8.75.2 Techniques

See 5.19.6.

8.75.3 Primary generated artefact-types

The following artefact-types will be used:

- GenericSpecification

8.76 Solution services and APIs nomenclature

8.76.1 General description

This is nomenclature of services and API of the solution. See 5.19.4.

This model-type is used in the following viewpoints:

- Solution engineering viewpoint (7.7)

8.76.2 Techniques

See 5.19.4.

8.76.3 Primary generated artefact-types

The following artefact-types will be used:

- Service
- API

8.77 Solution software factory configuration

8.77.1 General description

The platform software factory configuration describes all the details of an element (e.g. solution) related to its software development, testing, deployment and execution.

This model-type is used in the following viewpoints:

- Solution engineering viewpoint (7.7)

8.77.2 Techniques

The solution software factory configuration is based on the software factory viewpoint (7.11.

8.77.3 Primary generated artefact-types

The following artefact-types will be used:

- SoftwareConfiguration

8.78 Solution management and operations manual

8.78.1 General description

The solution management and operations manual provides the necessary information for the efficient evolution of the solution.

This model-type is used in the following viewpoints:

- Solution engineering viewpoint (7.7)

8.78.2 Techniques

The best organization of such a manual is a wiki-like structure and functionality.

8.78.3 Primary generated artefact-types

The following artefact-types will be used:

- WikiManual

8.79 Interoperability aspect

8.79.1 General description

Interoperability is the ability of systems to work together. The architecture principles of the solution space (see 5.19) help to address interoperability by-design requirement. The platform-based agile solutions orientation guarantees that smart cities which are built on the Common Smart Cities Digital Repeatable Platform will be interoperable because the platform defines all its interfaces (via APIs) and processes.

This model-type is used in the following viewpoints:

- Crosscutting aspects engineering viewpoint (7.8)

8.79.2 Techniques

A good framework of interoperability is given in ISO/IEC 21823-1:2019 [52].

8.79.3 Primary generated artefact-types

The following artefact-types will be used:

- API

8.80 Security aspect

8.80.1 General description

At present, the information security aspect is well-elaborated for infrastructure, data, information and applications. However, the information security aspect shall be also developed for business, namely for business-processes. This means that it should be clear how to place security points within digital processes. Also, an explicit link between security and risk shall be established (see 5.19.6).

All the security-related activities shall be within well-defined digital processes. Life cycle of security-related artefacts shall be fully controlled and formalized.

The security aspect covers the following:

- Information security practices, controls, points, and methods.
- Personal security practices, controls, points, and methods.
- Physical security practices, controls, points, and methods.

This model-type is used in the following viewpoints:

- Crosscutting aspects engineering viewpoint (7.8)

8.80.2 Techniques

There are many bodies of knowledge related to security: JTC 1/SC 27, ISO/IEC 27000 [1], NIST, MITRE, ITU SG17-C758 [29].

8.80.3 Primary generated artefact-types

The following artefact-types will be used:

- InformationSecurityPoint
- InformationSecurityControl
- InformationSecurityMethod
- PersonalSecurityPoint

- PersonalSecurityControl
- PersonalSecurityMethod
- PhysicalSecurityPoint
- PhysicalSecurityControl
- PhysicalSecurityMethod

8.81 Privacy aspect

8.81.1 General description

Privacy is the right of an entity (normally an individual or an organization), acting on its own behalf, to determine the degree to which the confidentiality of its private information is maintained.

Subclause 5.19.6 provides sufficient guidelines for managing privacy. All the privacy-related activities shall be within well-defined digital processes. Life cycle of privacy-related artefacts shall be fully controlled and formalized.

This model-type is used in the following viewpoints:

- Crosscutting aspects engineering viewpoint (7.8)

8.81.2 Techniques

A good framework of privacy is the General Data Protection Regulations (GDPR) [28], ITU SG17-C758 [29].

8.81.3 Primary generated artefact-types

The following artefact-types will be used:

- PrivacyPoint
- PrivacyControl
- PrivacyMethod
- PrivacyInformationClassification
- PrivacyRule

8.82 Safety aspect

8.82.1 General description

Safety is freedom from risk which is not tolerable. Subclause 5.19.6 provides sufficient guidelines for managing safety. All the safety-related activities shall be within well-defined digital processes. Life cycle of safety-related artefacts shall be fully controlled and formalized.

This model-type is used in the following viewpoints:

- Crosscutting aspects engineering viewpoint (7.8)

8.82.2 Techniques

A good framework of safety is given in ISO/IEC Guide 51 [14], ISO 12100 [53], ISO 13849-1 [54], IEC 61508 [55], and ITU SG17-C758 [29].

8.82.3 Primary generated artefact-types

The following artefact-types will be used:

- SafetyPoint
- SafetyControl
- SafetyMethod
- SafetyRule

8.83 Reliability and resilience management aspects

8.83.1 General description

Reliability is the ability to perform as required, without failure, for a given time interval, under given conditions. Resilience is the ability to absorb or avoid damage without suffering complete failure. Subclause 5.19.6 provides sufficient guidelines for managing reliability and resilience.

This model-type is used in the following viewpoints:

- Crosscutting aspects engineering viewpoint (7.8)

8.83.2 Techniques

A good framework of reliability and resilience is given in the ISO/IEC 25000 series, ISO/TC 292, ITU SG17-C758 [29].

8.83.3 Primary generated artefact-types

The following artefact-types will be used:

- GenericSpecification

8.84 Low cost of operations and time-to-market aspects

8.84.1 General description

Low-cost of operations and time-to-market relate to effectiveness and efficiency of the life cycle of the system-solution and its elements. The architecture principles of the solution space (see 5.19) and the software factory (7.11) address together these aspects by a combination of the following techniques:

a) use of microservices for which versioning is easy, and

b) assembling solutions from microservices via digital processes.

The power of this approach is that usually a process instance life cycle is much shorter than an application version life cycle; thus changes can be introduced faster.

This model-type is used in the following viewpoints:

- Crosscutting aspects engineering viewpoint (7.8)

8.84.2 Techniques

SOA [43] and BPM [56] are available bodies of knowledge.

8.84.3 Primary generated artefact-types

The following artefact-types will be used:

- GenericSpecification

8.85 Self-reference aspect

8.85.1 General description

Cities are self-referential systems. Such systems know about themselves. Various elements of a city system shall "know" that they are an integral part of the city system. This implies that whenever it is possible, various plans, maps, schemas, layout of a city shall be provided in digital presentation.

This model-type is used in the following viewpoints:

- Crosscutting aspects engineering viewpoint (7.8)

8.85.2 Primary generated artefact-types

The following artefact-types will be used:

- GenericSpecification

8.86 Organizational structure

8.86.1 General description

Organizational structure (or organigram or organization chart) is a map of functions and, sometimes, positions within an organization.

This model-type is used in the following viewpoints:

- Corporate viewpoint (7.9)

8.86.2 Techniques

One of the requirements for organizationcharts is avoidance of potential conflicts of interest.

8.86.3 Primary generated artefact-types

The following artefact-types will be used:

- OrganizationChart

8.87 Governance structure

8.87.1 General description

Governance ensures that:

- Stakeholder needs, conditions and options are evaluated to determine balanced, agreed-on enterprise objectives.
- Direction is set through prioritization and decision making.
- Performance and compliance are monitored against agreed-on direction and objectives.

Some organizations have a complex of governing bodies for some parts of the organization or some crosscutting activities (e.g. a project management office is a governance body for project management).

This model-type is used in the following viewpoints:

- Corporate viewpoint (7.9)

8.87.2 Techniques

CoBIT®[10] [57] is an available body of knowledge.

8.87.3 Primary generated artefact-types

The following artefact-types will be used:

- GovernanceBody

8.88 Project management

8.88.1 General description

Project management is essential for any organization right now; thus project management practices shall be explicitly defined.

This model-type is used in the following viewpoints:

- Corporate viewpoint (7.9)

8.88.2 Techniques

PMI [58] is an available body of knowledge.

8.88.3 Primary generated artefact-types

The following artefact-types will be used:

- WikiManual

8.89 Corporate manual

8.89.1 General description

Corporate manual is a comprehensive document which describes how an organization is carrying out its work. Sometimes, it is called quality manual.

This model-type is used in the following viewpoints:

- Corporate viewpoint (7.9)

8.89.2 Techniques

ISO 9000 is an available body of knowledge.

8.89.3 Primary generated artefact-types

The following artefact-types will be used:

- WikiManual

[10] CoBIT is a trademark of ISACA. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named.

8.90 Independent evaluation

8.90.1 General description

Independent evaluation in an organization, usually, has the following core objectives:

- to contribute to enhanced learning,
- to provide a basis for accountability, and
- to promote an evaluation culture within the organization.

The main deliverable of the independent evaluation is a set of lessons learnt.

This model-type is used in the following viewpoints:

- Corporate viewpoint (7.9)

8.90.2 Techniques

Recommended bodies of knowledge are UN agencies and international financial organizations.

8.90.3 Primary generated artefact-types

The following artefact-types will be used:

- WikiManual

8.91 Ethics, integrity and anti-corruption

8.91.1 General description

The primary function of the ethics office is to provide advice and guidance to staff members on matters of ethics and conduct of the staff members in the workplace. A commitment to integrity, anti-corruption and the highest ethical standards is fundamental to any organization. Thus, the ethics office is the anchor of a successful corporate ethics culture and ensures awareness and understanding of the organization’s core values and ethical standards.

This model-type is used in the following viewpoints:

- Corporate viewpoint (7.9)

8.91.2 Techniques

ISO 26000 is an available body of knowledge.

8.91.3 Primary generated artefact-types

The following artefact-types will be used:

- WikiManual

8.92 Environment and health

8.92.1 General description

Environment and health are very important for any organization to understand and improve.

This model-type is used in the following viewpoints:

- Corporate viewpoint (7.9)

8.92.2 Techniques

World Health Organization (WHO) is an available body of knowledge.

8.92.3 Primary generated artefact-types

The following artefact-types will be used:

- WikiManual

8.93 Risk management aspect

8.93.1 General description

Risk is a combination of the probability of occurrence of harm and the severity of that harm. At present, the risk management aspect is well-developed. However, risks shall be linked to other crosscutting aspects as explained in 5.19.6.

This model-type is used in the following viewpoints:

- Risk management viewpoint (7.10)

8.93.2 Techniques

There are many bodies of knowledge related to risk management: ISO/IEC 31000 [59], NIST, etc.

8.93.3 Primary generated artefact-types

The following artefact-types will be used:

- RiskPoint
- RiskControl
- RiskMethod

8.94 Compliance management aspect

8.94.1 General description

Compliance management is about ensuring that policies and procedures are followed in accordance with their setup. At present, the compliance management aspect is well-developed. However, compliance shall be linked to system elements as explained in 5.19.6.

This model-type is used in the following viewpoints:

- Risk management viewpoint (7.10)

8.94.2 Techniques

There are many bodies of knowledge related to compliance management: ISO/TC 309, etc.

8.94.3 Primary generated artefact-types

The following artefact-types will be used:

- CompliancePoint
- ComplianceControl
- ComplianceMethod

8.95 Regulatory management aspects

8.95.1 General description

Regulations specify and organize what that authorizing body feels is appropriate behaviour. Regulations are established by the federal, state, or local government or their appropriating agency. Often, regulations are written to implement the specifics of a particular law. Organizations shall know regulations which are applicable to their activities.

At present, the regulatory management aspect is well-developed. However, regulatory elements shall be linked to system elements as explained in 5.19.6.

This model-type is used in the following viewpoints:

- Risk management viewpoint (7.10)

8.95.2 Techniques

OECD [60] is an available body of knowledge.

8.95.3 Primary generated artefact-types

The following artefact-types will be used:

- RegulationPoint

8.96 Crime prevention and detection

8.96.1 General description

Crime (including fraud) is an important factor behind business failure. In fact, business crimes are both a legal and an economic issue for corporates. In the context of business law crime is related to the injury of public through business operations. But criminal law is different from civil law. At present, the crime prevention and detection is well-developed. However, crime prevention and detection activities shall be linked to system elements as explained in 5.19.6.

This model-type is used in the following viewpoints:

- Risk management viewpoint (7.10)

8.96.2 Techniques

UN [61] is an available body of knowledge.

8.96.3 Primary generated artefact-types

The following artefact-types will be used:

- GenericSpecification

8.97 Auditing

8.97.1 General description

Auditing is about planning, organizing, directing and controlling a broad, comprehensive programme of organization audits both internally and externally. At present, the auditing is well-developed. However, auditing activities shall be proactively linked to system elements as explained in 5.19.6.

This model-type is used in the following viewpoints:

- Risk management viewpoint (7.10)

8.97.2 Techniques

UN [61] is an available body of knowledge.

8.97.3 Primary generated artefact-types

The following artefact-types will be used:

- GenericSpecification

8.98 Independent investigation

8.98.1 General description

Independent investigation is about crime investigation of people who are responsible for crime prevention with an organization. At present, the independent investigation is well-developed. However, the independent investigation shall be proactively linked to system elements as explained in 5.19.6.

This model-type is used in the following viewpoints:

- Risk management viewpoint (7.10)

8.98.2 Techniques

UN [61] is an available body of knowledge.

8.98.3 Primary generated artefact-types

The following artefact-types will be used:

- GenericSpecification

8.99 Crisis management

8.99.1 General description

Crisis management is a set of processes by which an organization deals with a disruptive and unexpected event that threatens to harm the organization or its stakeholders. Such a set of processes addresses the following stages of a crisis:

- Preparation,
- Emergency response, and
- Recovery.

At present, the crisis management aspect is well-developed. However, crisis-related activities shall be linked to system elements as explained in 5.19.6.

This model-type is used in the following viewpoints:

- Risk management viewpoint (7.10)

8.99.2 Techniques

IFRC [62] and ICRC [63] are available bodies of knowledge.

8.99.3 Primary generated artefact-types

The following artefact-types will be used:

- WikiManual

8.100 Software factory overview

8.100.1 General description

A software factory is a software product line that configures extensive tools, processes, and content to automate the development and maintenance of digital elements of digital systems. Obviously, these digital elements are, predominantly, pieces of software. The software factory automates (thus eliminates from humans) all the non-creative activities from the traditional practices of application development and maintenance. The software factory is built in accordance with the selected:

- architecture of digital systems (how to decompose solution into units-of-work to be done),
- digital elements and systems life cycle (what units-of-work are ready to advance in their maturity), and
- work management practices (what units-of-work to be done by whom and when).

In theory, the life cycle of a digital element is rather straightforward:

- Business case phase.
- Architecting (or elaboration) phase.
- Construction (or build or implementation) phase.
- Transition (or deployment) phase.
- Production (or operating) phase.
- Retiring phase.
- Decommissioning phase.

However, in practice, the life cycle of a digital element is rather complex, because:

a) a digital element may comprise a few other digital elements (thus the architecting and engineering phases may be recursive);
b) automated testing and deployment into a non-production environment may be required for piloting;
c) a few versions of the same digital element may co-exist even in the production environment;
d) quick rework may be required because of some circumstances (errors, dependencies, etc.);
e) refactoring.

The software factory addresses this complexity by providing several practices, which can be combined into the above-mentioned phases: prototyping, engineering, assembling, testing, deployment and monitoring. Thus, it is possible to talk about BizDevOps instead of DevOps. Obviously, architecting is not mentioned because it is covered by the SCRAM and SCRA.

The software factory simplifies the configuration management which is the base of information security.

This model-type is used in the following viewpoints:

- Software factory viewpoint (7.11)

8.100.2 Techniques

An Association for All IT Architects [64], ITIL [50] and ITSM [51] are available bodies of knowledge.

8.100.3 Primary generated artefact-types

The following artefact-types will be used:

- SoftwareConfiguration

8.101 Prototyping practices

8.101.1 General description

Prototyping practices are used to collect and validate requirements from the users. Prototyping practices is based on the ability of some tools to interactively define the behaviour of some solution artefacts and enact it (e.g. UI forms, business rules, business processes, etc.).

This model-type is used in the following viewpoints:

- Software management viewpoint (7.11)

8.101.2 Techniques

Recommended bodies of knowledge are: PaaS-based tools.

8.101.3 Primary generated artefact-types

The following artefact-types will be used:

- GenericSpecification

8.102 Engineering practices

8.102.1 General description

Engineering practices are model-based software development with intensive use of domain-specific languages.

This model-type is used in the following viewpoints:

- Software management viewpoint (7.11)

8.102.2 Techniques

Model-Based Software Engineering (MBSE) is an available body of knowledge.

8.102.3 Primary generated artefact-types

The following artefact-types will be used:

- GenericSpecification

8.103 Assembling practices

8.103.1 General description

Assembling practices are necessary for handling numerous microservices.

This model-type is used in the following viewpoints:

- Software management viewpoint (7.11)

8.103.2 Techniques

Kubernetes®[11] [65] is an available body of knowledge.

8.103.3 Primary generated artefact-types

The following artefact-types will be used:

- GenericSpecification

8.104 Testing practices

8.104.1 General description

Well-known testing practices (unit, integration, system and performance) are extended by resilience test.

This model-type is used in the following viewpoints:

- Software management viewpoint (7.11)

8.104.2 Techniques

[66] is an available body of knowledge.

8.104.3 Primary generated artefact-types

The following artefact-types will be used:

- GenericSpecification

8.105 Deployment practices

8.105.1 General description

Deployment practices shall be adapted to handle containers and multi-clouds.

This model-type is used in the following viewpoints:

- Software management viewpoint (7.11)

8.105.2 Techniques

Docker [67] is an available body of knowledge.

8.105.3 Primary generated artefact-types

The following artefact-types will be used:

- GenericSpecification

8.106 Monitoring practices

8.106.1 General description

Monitoring practices shall be adapted to handle various temporary environments (ideally, one change may have one or more environments).

[11] Kubernetes is a trademark of the Linux Foundation. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named.

This model-type is used in the following viewpoints:

- Software management viewpoint (7.11)

8.106.2 Techniques

Nagios®[12] is an available body of knowledge (see https://www.nagios.org/).

8.106.3 Primary generated artefact-types

The following artefact-types will be used:

- GenericSpecification

8.107 Existing standards nomenclature

8.107.1 General description

The SCRA provides the structure and several classifications for the system-solution which should be used for tagging existing standards.

This model-type is used in the following viewpoints:

- Standards viewpoint (7.12)

8.107.2 Techniques

Recommended bodies of knowledge are: IEC, JTC 1, BSI, DIN, SF-SSCC (Sector Forum on Smart and Sustainable Cities and Communities, a joint activity of CEN, CENELEC and ETSI).

8.107.3 Primary generated artefact-types

The following artefact-types will be used: will be defined in IEC 63233 [68].

8.108 Potential standards nomenclature

8.108.1 General description

The SCRA provides the structure of the system-solution and related interfaces between its elements which should be used for defining potential standards.

This model-type is used in the following viewpoints:

- Standards viewpoint (7.12)

8.108.2 Techniques

Recommended bodies of knowledge are: IEC, JTC 1, BSI, DIN, SF-SSCC (Sector Forum on Smart and Sustainable Cities and Communities, a joint activity of CEN, CENELEC and ETSI).

8.108.3 Primary generated artefact-types

The following artefact-types will be used: will be defined in IEC 63233 [68].

[12] Nagios is a trademark of Nagios Enterprises. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the product named.

Annex A
(informative)

Viewpoints from the Zachman Framework

The Zachman Framework [30] is an enterprise ontology and is a fundamental structure for Enterprise Architecture which provides a formal and structured way of viewing and defining an enterprise. The ontology is a two-dimensional classification schema that reflects the intersection between two historical classifications. The first are primitive interrogatives: What, How, When, Who, Where, and Why. The second is derived from the philosophical concept of reification, the transformation of an abstract idea into an instantiation. The Zachman Framework reification transformations are: Identification, Definition, Representation, Specification, Configuration and Instantiation [69].

The Zachman Framework typically is depicted as a bounded 6 × 6 "matrix" with the Communication Interrogatives as Columns and the Reification Transformations as Rows. The framework classifications are repressed by the Cells, that is, the intersection between the Interrogatives and the Transformations.

Because of this universal nature of the Zachman Framework, it can be used for creating some practical viewpoints and model-kinds. Below, six viewpoints and six model-kinds per viewpoint are defined. Selection of a particular modelling technique for a model-kind is free.

The descriptions are taken directly from version 3.0 of the Zachman Framework.

Executive viewpoint

1) (What) Inventory Identification model-kind
2) (How) Process Identification model-kind
3) (Where) Distribution Identification model-kind
4) (Who) Responsibility Identification model-kind
5) (When) Timing Identification model-kind
6) (Why) Motivation Identification model-kind

Business Management viewpoint

1) (What) Inventory Definition model-kind
2) (How) Process Definition model-kind
3) (Where) Distribution Definition model-kind
4) (Who) Responsibility Definition model-kind
5) (When) Timing Definition model-kind
6) (Why) Motivation Definition model-kind

Architect viewpoint

1) (What) Inventory Representation model-kind
2) (How) Process Representation model-kind
3) (Where) Distribution Representation model-kind
4) (Who) Responsibility Representation model-kind
5) (When) Timing Representation model-kind
6) (Why) Motivation Representation model-kind

Engineer viewpoint

1) (What) Inventory Specification model-kind
2) (How) Process Specification model-kind
3) (Where) Distribution Specification model-kind
4) (Who) Responsibility Specification model-kind
5) (When) Timing Specification model-kind
6) (Why) Motivation Specification model-kind

Technician viewpoint

1) (What) Inventory Configuration model-kind
2) (How) Process Configuration model-kind
3) (Where) Distribution Configuration model-kind
4) (Who) Responsibility Configuration model-kind
5) (When) Timing Configuration model-kind
6) (Why) Motivation Configuration model-kind

Enterprise viewpoint

1) (What) Inventory Instantiations model-kind
2) (How) Process Instantiations model-kind
3) (Where) Distribution Instantiations model-kind
4) (Who) Responsibility Instantiations model-kind
5) (When) Timing Instantiations model-kind
6) (Why) Motivation Instantiations model-kind

Annex B
(informative)

Summary of the SRG work on systems approach

B.1 Systems approach foundation

The systems approach is a holistic, iterative, discovery process that helps with first defining the right problem in complex situations and then with finding elegant, well-designed and working solutions. It incorporates not only engineering, but also logical human and social aspects.

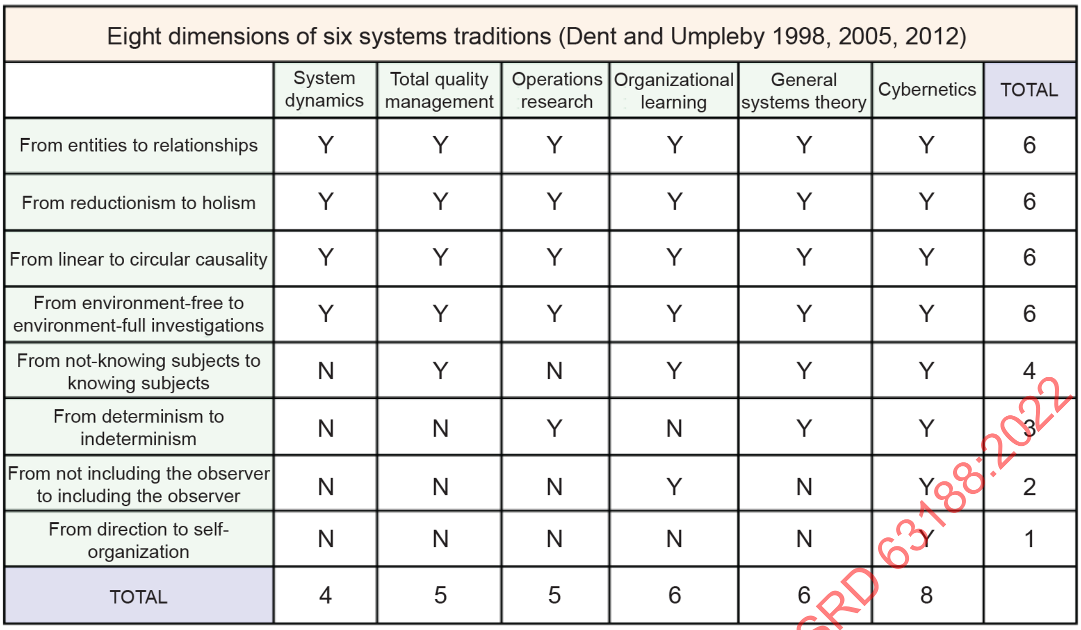
Use of the systems approach makes explicit the structure of a system with the rules governing its behaviour and helps understanding the relations the elements and parts of the system have among themselves and with the environment.

The need for a systems approach derives from the fact that nowadays it is increasingly recurrent that functional and structural engineering, system-wide interfaces and desired system properties become more and more important due to the increasing complexity, convergence and interrelationship of technologies.

The goal of any systems approach is to walk people and organizations working on complex systems through various stages and steps of analysis and synthesis in order to achieve the following:

- build a comprehensive understanding of the problem space (i.e. abstract domain in which the problem resides) including a list of people and organizations interested in potential solutions (i.e. stakeholders);
- outline a set of essential characteristics of the solution space (i.e. abstract domain in which potential solutions to the problem reside);
- architect and engineer a potential solution (i.e. the particular system-of-interest) at any desired level of detail.

There are several "systems traditions" as shown in Figure B.1. The IEC SRG has been developing a systems approach that aims at covering all the mentioned dimensions.



Eight dimensions of six systems traditions (Dent and Umpleby 1998, 2005, 2012)

	System dynamics	Total quality management	Operations research	Organizational learning	General systems theory	Cybernetics	TOTAL
From entities to relationships	Y	Y	Y	Y	Y	Y	6
From reductionism to holism	Y	Y	Y	Y	Y	Y	6
From linear to circular causality	Y	Y	Y	Y	Y	Y	6
From environment-free to environment-full investigations	Y	Y	Y	Y	Y	Y	6
From not-knowing subjects to knowing subjects	N	Y	N	Y	Y	Y	4
From determinism to indeterminism	N	N	Y	N	Y	Y	3
From not including the observer to including the observer	N	N	N	Y	N	Y	2
From direction to self-organization	N	N	N	N	N	Y	1
TOTAL	4	5	5	6	6	8	

IEC

Figure B.1 – Systems traditions

B.2 Structure of the SRG work on systems approach

Originally, work on systems approach in the IEC was carried out by the Systems Resource Group (SRG) as a group under the IEC Standardization Management Board. In the year 2021, SRG was disbanded and its activities were transferred to Strategic Group 12, which is a group under the IEC Standardization Management Board. Although the SRG didn't publish any final recommendations, its work is foundational for the SCRAM and this document refers to that foundation as "summary of the SRG work on systems approach ".

The summary of the SRG work on systems approach was aimed at a distinct standardization purpose: to find opportunities for standardization. To achieve this, in the most complex case it is necessary to develop the system domain reference architecture and, partially, the system domain reference solution architecture.

The summary of the SRG work on systems approach is a guideline that provides a systematic approach with tools and methodologies for committees working on complex systems. The intent of the approach is to walk users through various stages of analysis in order to build a comprehensive understanding of the system, scope, and ultimately be able to identify gaps where standards are needed. The summary of the SRG work on systems approach can be applied to any system of interest at any desired level of detail depth.

There are six stages in the summary of the SRG work on systems approach. Each stage is focused on an area of understanding about the system. The knowledge gained from each stage builds on top of the previous stages, therefore it is important to follow each stage in order. However, users are encouraged to apply an iterative approach, meaning there is no constraint on going back to a previous stage with new insights that were gathered in another stage. These "iterative" practices are described below.

STAGE 1 – Domain of Interest Analysis
This initial stage is crucial to analyse and identify the market/industry trends and needs. The outputs of this stage set the foundation, scope, and boundaries of the system of interest.

STAGE 2 – Stakeholders Analysis
The second stage extrapolates on the first with a focus on identifying the stakeholders and defining their needs, objectives and concerns.

STAGE 3 – Use Case Analysis
Stage three is focused primarily on identifying use cases and on developing a progressively detailed understanding of the use cases identified. The use case methodology in IEC 62559-2 [40] can be used for the purpose. Through the Use Case Analysis, System Requirements can be derived, both functional and non-functional.

STAGE 4 – Systems Architecting and Modelling
Through this stage, the architecture of the system is defined with its structure and behaviour, i.e. what are its parts and how do these parts interact, to directly respond to the systems requirements previously defined. This exercise can continue, if needed, to progressively model and design in detail system discrete parts, interfaces and flows to help build a holistic perspective of the system.

STAGE 5 – Standards Analysis
In this stage, the focus is on understanding and mapping what relevant standards exist for all the various parts of the system and whether they are contributing to or countering the objectives identified in Stage 1. These standards can include IEC and other SDO standards.

STAGE 6 – Gap Analysis
In the final stage, gaps where standards are missing are identified based on the knowledge of existing standards, desired system interaction, use cases and other information gathered from the previous stages. This will then initiate the activities for the committees to move forward with new standard development.

Schematically, the summary of the SRG work on systems approach is shown in Figure B.2.

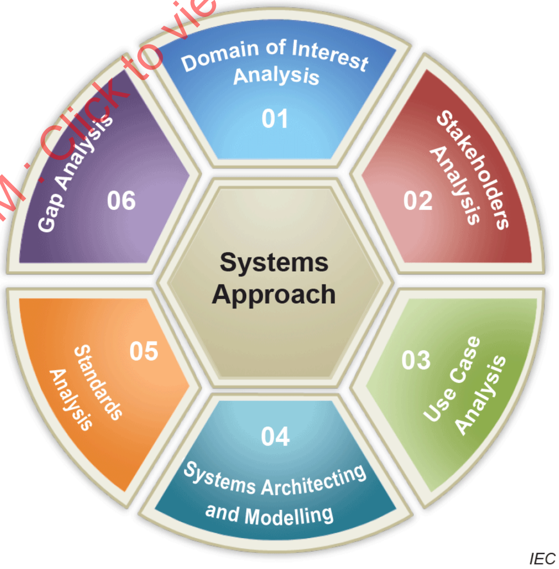


Figure B.2 – The summary of the SRG work on systems approach

B.3 ISO/IEC/IEEE 42010 foundation

The summary of the SRG work on systems approach stage of Systems Architecting can refer to ISO/IEC/IEEE 42010:2011. To describe an architecture of a system, this standard uses the following logic and the several digital work products:

- the system-of-interest architecture is communicated by its architecture description;
- architecture description is a collection of several architecture views;
- each architecture view is a collection of architecture models to address of some specific system-of-interest concerns.

For the needs of the summary of the SRG work on systems approach, this logic has been extended by the following:

- each architecture model formally codifies some relationships between some architecture artefacts;
- an architecture artefact is an entity made by creative human work; and
- nomenclature is a classification of various architecture artefacts of the same type (the same artefact may appear in more than one architecture model).

Architecture views, architecture models and architecture artefacts depend on the system-of-interest.

To facilitate production of architecture views and architecture models, ISO/IEC/IEEE 42010:2011 proposes to use two extra work products which are systems-of-interest independent:

- architecture viewpoint to govern the construction, interpretation and use of architecture views; and
- model-kind to govern the construction, interpretation and use of architecture models.

For the needs of the summary of SRG work on systems approach, this logic has been extended by the following:

- artefact-types to govern the construction, interpretation and use of architecture artefacts; and
- patterns to facilitate creation of some model-kinds from other model-kinds.

Many viewpoints and views are possible as shown in Figure B.3.

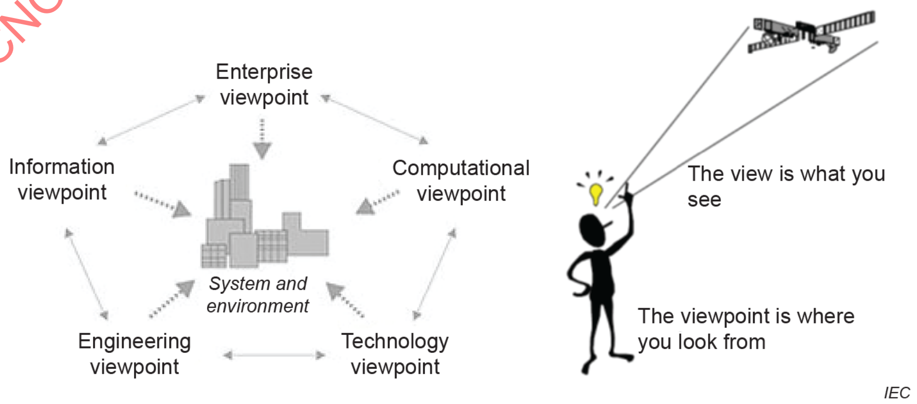


Figure B.3 – Viewpoints and views

Different stakeholders see the same system-of-interest differently and recognize different artefacts as shown in Figure B.4.

IEC

Figure B.4 – Different views, models and artefacts

B.4 Systems architecting foundation

B.4.1 General

The concept "reference architecture" is one of the central concepts in the summary of the SRG work on systems approach and several related digital work products are used:

- SRG systems approach terminology to explain various concepts of the systems approach and relationships between them;
- (optionally) System Domain Reference Architecture Methodology – any adaptation of the summary of the SRG work on systems approach for a particular system domain.
- System Domain Reference Model – the key concepts and relationships between them in a particular system domain.
- System Domain Reference Architecture – the key decisions taken to guarantee that future systems (built in accordance with this reference architecture) will possess the required essential characteristics, e.g. security and privacy by-design.
- (optionally) System Domain Reference Solution Architecture – an adaptation of the System Domain Reference Architecture and its deeper engineering for validating or standardization purposes. In other words, reference solution architecture is a detailization of a System Domain Reference Architecture. This may be necessary to find components and interfaces between them to be standardized.
- (optionally) System Domain Reference Implementation – a realization of the System Domain Reference Solution Architecture for validating purposes. Also known as "testbed".

It is considered that a client wants to use the System Domain Reference Architecture to create the following digital work products:

- Solution Architecture – any adaptation of the System Domain Reference Architecture for a particular case. Also known as system architecture or blueprint.
- Implementation – a realization of a solution.

Typical scenarios for SyC Smart Cities work are the following.

B.4.2 Simple scenario

Only a "Reference Architecture" is developed as shown in Figure B.5. The arrows between various figures are interpreted in the following way:

- "Various needs" are used to extract some concepts which are relevant for the problem space and to build from them "Reference model" as a terminological concept system. While "Reference model" is matured, "Various needs" may be aligned to reflect "Reference model" (e.g. remove duplicative concepts).
- "Various needs" and "Reference model" are used to architect "Reference architecture". While "Reference architecture" is maturing, some new essential concepts or relationships may be added into "Reference model".
- "Reference architecture" (as a template) is used to architecture and engineer a particular "Solution architecture". The experience with "Solution architecture" may be used to refine "Reference architecture".
- "Solution architecture" is used to build "Implementation". Again, the experience with "Implementation" may be used to refine "Solution architecture".

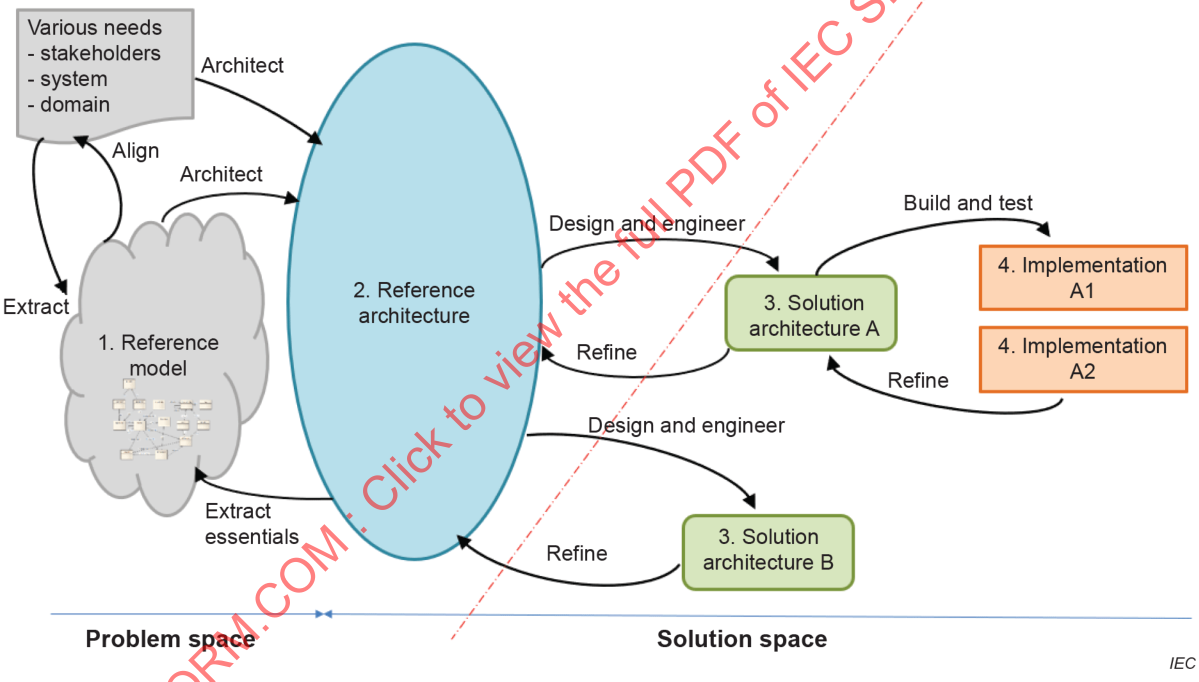


Figure B.5 – Simple scenario

B.4.3 Average scenario

As shown in Figure B.6, in this scenario, a "Reference Solution Architecture" is developed to find out more detailed reference capabilities and interfaces and refine the "Reference Architecture". This "Solution Reference Architecture" may be additionally tested by a "Reference Implementation".