ITUPublications

International Telecommunication Union
Telecommunication Standardization Bureau

Contribution SG 17-C3000

(01/2025)

Design Principles and Best Practices for Security Architectures

CAUTION! CONTRIBUTION

This is an ITU Contribution. It is an internal document to ITU, and it is not to be used outside of ITU.

Contribution SG 17-C3000

Design Principles and Best Practices for Security Architectures

INTERNATIONAL TELECOMMUNICATION UNION

TELECOMMUNICATION STANDARDIZATION BUREAU OF ITU

SG 17-C3000

STUDY GROUP 17

Original: English

STUDY PERIOD 2025–2028

Question(s): 1/17 TODO-PLACE, 01 Feb 2025/02 Feb 2025

CONTRIBUTION

Source: Broadcom Europe Ltd.

Title: Design Principles and Best Practices for Security Architectures

Contact: Arnaud Taddei Tel. +41795061129
Broadcom Europe Ltd. E-mail Arnaud.Taddei@broadcom.com
United Kingdom

Contact: Tel.

Abstract: This contribution contains a proposed revised baseline text for X.arch-design

Introduction

This contribution provides a new proposed revised baseline text for X.arch-design. It took into consideration all the former editor notes and recognized a particular big obstacle in e.g. the need to define Zero Trust as an ITU-T wide definition.

Keywords

Architect, Design, Designer, Design Principle, Security.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications , information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

CAUTION! CONTRIBUTION

This is an ITU Contribution. It is an internal document to ITU, and it is not to be used outside of ITU.

1. Proposal

It is proposed that Q1 reviews this contribution, discussed it and agrees how to progress X.arch-design baseline text.

EDITORIAL NOTE

- (Former editors notes below)
- This document contains the revised baseline text for X.arch-design based on C821 with no change
- These editors notes are now regrouped from 3 sources
  - The initial editors notes from C821
  - All the contribution C801 as a block
  - And one key remark during Q1 meeting that to resolve the issue paused by Zero Trust definition, it is proposed to use the language: "working definition for the purpose of this definition/recommendation"
- Editors note the good compatibility of all of these contributions with no clashes and therefore will execute step by step all the proposals as there is not enough time in this SG17 meeting to do a good quality-controlled job
- Editors note too that this above work will require a non-neglectable amount of time.

All editors notes from C821 and C803 kept verbatim for further execution below:

Summary Inputs:

This contribution provides some inputs for making the X.arch-design crisper and more coherent with other related documents under development in SG17 and specifically in SG 17/Q1. The baseline Text has NOT been edited or modified to enable the editors to undertake this exercise in their own structure based on their acceptance of the inputs.

a) The Proposed Title for the recommendation: X.arch-design: Security Design Principles and Concepts
b) The Recommendation should focus on primarily Two aspects of Security — Concepts & Principles.
c) The security concepts and principles laid out in this recommendation should be targeted to become the foundational cornerstones for all other Security related collaterals within SG 17/Q1 and even all other Questions in SG 17 or any other SG…
d) The security concepts and principles when applied to different stakeholders, application domains and use cases in Digital Systems, Solutions, Networks & Infrastructures shall enable insights to capture respective Security Concerns.
e) The captured Security Concerns shall enable developing the relevant Architecture Models, Views and Viewpoints.
f) A comprehensive collection of Security Concerns and their respective Architecture Models, Views & Viewpoints shall enable composing the wholistic Cyber Security Reference Architecture.
g) It is proposed that this recommendation along with TR.cs-uc, TR.cs-sc and X.cs-ra should be comprehensively coherent and complementary among themselves and there should be NO overlapping of same information.
h) As already acknowledged, it shall be appropriate to allocate the extra content to the relevant document(s) under development.
i) The graphics below illustrates the most optimum relationship and coherence amongst all the documents under development in SG 17/Q1.

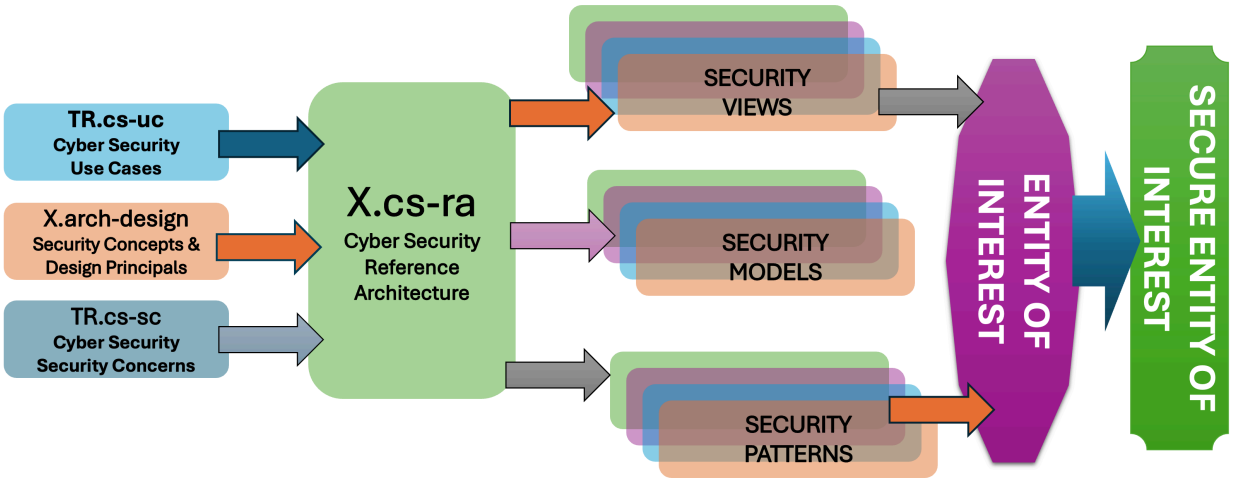

Figure 1 — Relationship amongst SG 17/Q1 deliverables

A few key security Principles are submitted for consideration of the editors for inclusion with appropriate context:

Key Security Principles

- Security characteristics must be considered holistically.
- A culture of Security is essential to achieving Security.
- Applicability of t Security to any entity — Security can be defined, verified, validated, and demonstrated for an entity.
- Security is contexts-dependent — Security of an entity depends on its environment(s) and context of use. Understanding context is necessary for making Security trade-offs.
- Security should apply risk management — It should be possible to define, validate and verify the level of trust required for a specific system by using a risk management approach (e.g. ISO/IEC 31000). Security risks and requirements are addressed using Security Controls.
- Appropriate investment for Security — The costs of applying Security Controls and performing verification measurements should be commensurate with the unmitigated risk impacts cost.
- Security should be required — Security risks and requirements should be identified, defined, analyzed, and evaluated for each Security characteristics and for every stage of the entity's life cycle.
- Security must be managed throughout the entire system lifecycle.
- Security shall be demonstrable — The trusted entity auditing process makes use of the measurable, verifiable, and demonstrable evidence. Assurance based on evidence is essential to establish Security. Assurance requires a systems viewpoint with evidence of multiple factors.
- Security shall be transparent — The environments and contexts, Security characteristics, processes, stakeholders, information, risks, controls, outcomes, and evidence related to a declaration of Security in an entity shall be clearly defined, approved by stakeholders, and communicated.

Introduction

Since the last SG17 meeting a lot of team work and background work was performed between several SG17 members that lead to a context that had to be taken into account for this Contribution.

Indeed:

– the initialisation and development of the work in X.cs-ra,
– the welcome new work item proposal C583,
– the Broadcom proposal for a CRAMM Roadmap internal SG17 document as per C652,
– the outcome and the future of CG-SECAPA,
– the fact that some work items in the incubation queue of Q15 may be claimed by Q1,

is creating a context that led the contribution to decide to purge any aspect which is on developing reference architectures and methodology from this document in order to limit the contour of this contribution to solely on:

What are the design security principles and associated that can constitute an inventory of use for the architect/design for when preparing a reference architecture, a solution architecture or an implementation.

This will give the opportunity for a better delineation and focus of this document back to its origin but now helped by a bigger context and will allow this document to reuse results from other works if necessary rather than defining terminology outside of its scope.

The future CRAMM Roadmap may actually consider if the section on the Architect/Designer shall stay in X.arch-design or should be moved into a more wholistic document listing all the stakeholders that participate to a Reference Architecture, yet, acknowledging that the Architect/ Designer is a central one.

Therefore this contribution is progressing the work on X.arch-design in the following ways:

Remove the methodological section 8

The methodology should not be defined in this Recommendation.

Removal of most ISO and IEC references and terminology

Most of ISO and IEC references are useful but for methodology and more for X.cs-ra and other documents. Some key elements were kept.

The contributor is wondering if, as part of this nascent new series, a document regrouping all the references, terminology, etc. shouldn't be developed as a common denominator to avoid having to carry references and increase disalignment as much as the work is progressing.

Introduce a new section 6.2

The goal of this section is to give a reminder of what is done in other works to give a context and anchor the different work items. It may be tuned and refined in the future but the most important is that it shows that this Recommendation focuses on the point 3) mostly. This gives a clearer 'interface' between work items.

Regroup all the terms 'defined here' in the right place

This is in accordance to the document, yet a number of issues need to be considered:

– it is surprising to the editors that a number of these terms do not seem to be defined anywhere and more research is needed in databases and other SDOs,
– a number of interpretations need to be researched,
– keep investigating the right SG17 series of Recommendations and in particular SDL,
– a number of terms may need to be defined in other current and future Recommendations and may need to be removed from this document.

Make a convention section:

– this is to create labels and identifiers for normative references and easier identification and use by the users of this Recommendation.

Developed of a number of principles

– principle of least privilege,

– principle of Zero Trust,

Others

– the reference to RFC9413 is extremely relevant as illustrative of the complexity and wisdom that an architect/designer will need to exercise and was placed as an example in 6.3,
– the important remark on 'Despite the fact that security of some elements in the system can be proved, there is no definite way to measure and compare security of the whole system' is now included and helps to define the critical 'Juvenal' security design constraint that the contributor failed to find a good way to introduce it, now done,

To be done (as to not forget a number of useful considerations)

– keep developing all the sections,
– position Confidentiality, Integrity, Availability somewhere,
– confront this work with X.800,
– clarify 'cyber security' vs 'cybersecurity'. The focus is on security of the entity of interest which may not be just a 'cyber' entity of interest and it provides a much more powerful context when 'security' is taken from the perspective of a design characteristic vs others and in particular vs the 'dependability' and later 'resiliency work',
– is 'entity of interest' not a definition?
– re-read the CISA TIC document and extract the 'capabilities' that are in fact principles,
– rework completely section 8 on architect/designer and before anything discuss if it should stay in this Recommendation or if a new work item should be proposed to regroup the definitions of all the stakeholders in which the architect and the designer are central but not alone,
– add about 'beyond corp' and then 'Jericho forum',
– consider the concept of 'architecture building block ABB' from the opengroup and see their ZT architecture document (see in CG-SECAPA),
– consider developing this one with metanorma and under GitHub. That would ease significantly maintaining the list of principles in proper tables, cross references, etc. and avoid mistakes.

The following todo list is kept here from the editorial notes of the current baseline text for the record

In this contribution:

– a few minor editorial notes were introduced,
– Confidentiality, Integrity and Availability are not security design principles, they are security properties,
  • There are no definitions for the term 'security properties' in the ITU terms and definitions database. Security properties could be interpreted as an architectural characteristic and as per CG-SECAPA work should not be positioned in this Recommendation,
– X.800 is being re-studied and X.800 makes sense vs X.200,
  • It is necessary to reanalyse X.800 vs this Recommendation and identify concepts and security design principles if any though it is likely that X.800 will probably be a better input in the architecture parts of CRAMM as per CG-SECAPA,
– the definition of Zero Trust was further developed but it shows a lot of issues:
  • The initial Forrester document from 2010 couldn't be found on internet at this stage and seems super-seeded by many newer documents that have a paywall
  • The NIST definition doesn't seem to be a definition as per ITU-T authors guide annex B.3.2 and so DEF01 is kept as-is,
  • The variations of Zero Trust semantic are absolutely huge making it very difficult to capture any alternatives for the reader,
  • Need to introduce a MECE approach

- Observe that X.ztmc will have similar issues

– BeyondCorp main documents were identified down to 2014 but not down to 2009,
– CG-SECAPA focus on CRAMM shows a number of issues about this contribution that need to be discussed

Proposal

– there is a critical need to have a work shop on the topic with all concerned experts as there are some clear issues to identify and define a number of foundational terms.

ITUPublications

International Telecommunication Union
Telecommunication Standardization Bureau

Recommendation

Provisional identifier: X.arch-design
(01/2025)

SERIES X: Data networks, open system communications and security

Recommendations

Design Principles and Best Practices for Security Architectures

Recommendation

Design Principles and Best Practices for Security Architectures

Summary

The summary goes here.

Keywords

Architect, Design, Designer, Design Principle, Security.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications , information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Page

Design Principles and Best Practices for Security Architectures

1. Scope

The scope of this recommendation is the definition of a lightweight, pragmatic and proven set of design principles, concepts, and criteria; and how to select and apply them to any security design or architectural work.

2. References

None.

3. Definitions

EDITORIAL NOTE

To Be Verified towards the end of the development of this Work Item if we need all of these definitions as we focused the scope on basic concepts and principles

3.1. Terms defined elsewhere

None.

3.2. Terms defined elsewhere

3.2.1. concern: [ISO/IEC/IEEE 42010:2022], Clause 3.7: <system> interest in a system relevant to one or more of its stakeholders

NOTE – 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

3.2.2. entity of interest: [ISO/IEC/IEEE 42010:2022], Clause 5.2.1: The term "entity of interest" is used in this document to refer to the subject of an architecture description. The term is intended to encompass, but is not limited to, entities within the following fields of application, reflecting the intended scope of this document as specified in clause 1.

- software, including software products and services, per ISO/IEC/IEEE 12207;
- systems, including one-of-a-kind systems, mass-produced systems, customized, adaptive systems, stand-alone and embedded systems, per ISO/IEC/IEEE 15288;
- enterprises as described in ISO 15704, i.e. human undertakings or ventures that have mission, goals and objectives to offer products or services, or to achieve a desired project outcome or business outcome.

3.3. Terms defined in this recommendation

This Recommendation defines the following terms:

3.3.1. architecture:

EDITORIAL NOTE

should be defined elsewhere

NOTE – an architecture identifies a particular problem space and defines a technology-independent analysis of requirements.

3.3.2. characteristic: a property of a system of interest.

EDITORIAL NOTE
should be defined elsewhere

3.3.3. design:

EDITORIAL NOTE
should be defined elsewhere

NOTE – a design maps architectural requirements into a particular family of solutions based upon standards and technical approaches.

3.3.4. framework:

EDITORIAL NOTE
should be defined elsewhere

NOTE – a framework sits at a broad, conceptual level and provides context for more detailed technical aspects.

3.3.5. implementation: realisation of an entity of interest.

EDITORIAL NOTE
should be defined elsewhere

3.3.6. reference architecture: template for solution architecures which realizes a prefefined set of requirements.

NOTE – A reference architecture uses its subject field reference model (as the next higher level of abstraction) and provides a common (architectural) vision, a modularization and the logic behind the architectural decisions taken.

3.3.7. reference model: abstract framework for understanding concepts and relationships between them in a particular problem space (or subject field).

3.3.8. security architectural principle: a guiding believe or rule that informs the design and development of the security aspects within an architecture.

3.3.9. security concern: interest to the security aspects of an entity of interest relevant to one or more of its stakeholders.

NOTE 1 – The same NOTE as for the term concern in section 3.1 applies: 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.

NOTE 2 – These concerns encompass the identification and understanding of potential security risks, vulnerabilities, threats and protective measures that need to be addressed within the architecture.

3.3.10. security design: the process of conceptualizing, selecting, tailoring and organizing the composition of the appropriate security capabilities and security design principles to protect a specific entity of interest throughout its lifecycle.

NOTE – this involves assessing risks, identifying security concerns, security requirements and applying relevant security design principles - such as Zero Trust, Defense in Depth, and the Principle of Least Privilege - to develop the corresponding architecture (reference, solution, implementation)

3.3.11. security design best practice: The established and proven techniques, methodologies, and guidelines that represent the most effective and reliable approaches for enhancing the security of a specific entity of interest.

3.3.12. security design consideration: the factors that influence the security design for a specific entity of interest.

3.3.13. security design constraint: a limitation or requirement that shapes the selection, organization and implementation of security capabilities and security design principles within the security design process.

NOTE – these constraints can stem from regulatory requirements, technical limitations, business objectives, or environmental factors, and they directly influence the development of security architectures and solutions to ensure protection of a specific entity of interest throughout its lifecycle.

3.3.14. security design principle: a guiding believe or rule that directs the security design of an entity of interest.

3.3.15. security meta reference architecture framework: a higher-level framework that provides a structured approach for creating reference architectures within the security domain knowledge. It defines the common components, models, principles, and best practices that can be applied across various reference architectures.

3.3.16. solution: should be defined elsewhere

EDITORIAL NOTE
should be defined elsewhere

NOTE – a solution manifests a design into a particular vendor technology, ensuring adherence to designs, models, and frameworks.

3.3.17. solution architecture: architecture of an entity of interest.

NOTE – a solution architecture (also known as a blueprint) can be a tailored version of a particular reference architecture (which is the next higher level of abstraction).

4. Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

DEF	DEFinition
MECE	Mutually Exclusive Collectively Exhaustive

PoLP	Principle of List Privilege
SAP	Security Architecture Principle
SDB	Security Design Best Practice
SDC	Security Design Consideration
SDP	Security Design Principle
SDX	Security Design Constraint

5. Conventions

In this document the following conventions will be used.

The label DEFxx is labelling a definition in a given principle.

The label SAPxx is labelling a security architecture principle.

The label SDPxx is labelling a security design principle.

Labels can be combined into identifiers in an absolute name space.

EXAMPLE

SDPxxDEFyy is the identifier which represents the definition yy in the security design principle xx.

6. Context

6.1. Introduction

The audience of this Recommendation is a designer and/or an architect in need to produce a security reference architecture, or a derived security solution architecture, or the derived actual security solution implementation and lifecycle.

Whilst security is an imperative for any design, security is only one aspect of the overall design and, in this perspective, security is only one characteristic among a growing number of conflicting characteristics.

Achieving security within a design requires the support of a number of meta-reference architecture elements and this Recommendation will concentrate on:
– Design principles and best practices for security architectures (clause 7),
– An understanding of the role of the designer and/or architect (clause 9).

6.2. Architectural methodological reminders

The adoption of architecture practice is a strategic decision for an organization that can help improve its overall value to a variety of stakeholders. Developing and using architectures in any domain has major benefits because a well-developed architecture can:
– foster stakeholder engagement and cooperation with decision-making activities,
– promote uniformity of products and services delivered,
– frame development and usage of solutions (including products, services and systems),
– increase the efficiency and effectiveness of transformation or modernization initiatives,
– promote coherence between enterprise and technical solutions (e.g. systems, software, services),
– improve interoperability between enterprises, systems, services and software applications,
– improve compatibility between systems and technologies,
– drive development of technologies for future applications,

– provide a framework for identifying teams and enabling systems,
– meet consumer demand in the evolving landscape of the marketplace,
– help structure a plan and integration points.

The architecting principles are defined in three categories:

a) Principles about the meaning of architecture
   1) architecture as embodiment of decisions
   2) understanding the problem space and solution space
   3) identifying fundamental concepts and properties of the architecture
   4) architecture as abstractions relevant to nature of architecture entity
b) Principles about the intent of architecture
   1) architecting with a focus on informing decision making
   2) architecting with a focus on value
   3) achieving a balanced and robust architecture
   4) describing architecture to enhance understanding of its intent
c) Principles about the nature of architecture
   1) architecting with a focus on key architectural properties
   2) architecting with a focus on relationships and interfaces
   3) identifying principles guiding solution development
   4) identifying principles guiding the evolution of architecture entities

This Recommendation focuses on c) and in particular c) 3) and c) 4) though the reader of this Recommendation should be mindful of the wider context of this Recommendation.

7. Concepts

7.1. The complex and nuanced nature of the object called 'Security'

In the context of this Recommendation, on a technological level only, security architectures can be interpreted as:
– security is an architecture,
– security is a design and/or architecture characteristic,
– security is a design and/or architecture criterion,
– security needs to follow some set of design principles for architecture,

When considering an entity of interest, all the security measures form an architecture by themselves and all the above interpretations should be considered at the same time.

This security architecture:
– like any, is subject to comply to a number of functional and non-functional characteristics,
– is therefore subject to the security characteristics itself,
– needs to follow some set of design principles for architecture,
– may be evaluated against various criteria including security criteria.

This approach is partly revealing one aspect of the significant complexity of the nature of security architecture on a technological level only.

It should be completed with the fundamental issue that despite the fact that security of some elements in the system can be proved, there is no definite way to measure and compare security of the whole system.

This will be called the Juvenal security design constraint in reference to the famous quote: 'sed quis custodiet, ipsos custodet' which can be interpreted as 'who guards the guards'. This security design constraint represents a key 'glass roof' that may be pushed, may be deformed but doesn't seem to have any possibility to be pierced.

All the above considerations are part of an even wider context. Indeed, the theory of design includes three other dimensions of law, ethics and anthropology that the architect and/or designer needs to consider when developing a design. Whilst this is clearly important, these dimensions are not in the scope of this Recommendation, yet they are represented as attributes in the role of the architect and/or designer in this document.

Whilst there are few well-constructed examples to illustrate the logical complexity that the above represents, a good example can be found in [b-RFC9413] in a specific context of the Maintaining Robust Protocols. Robustness is a typical example of a design characteristic that is expressed and it shows how this 'robustness principle' led to unanticipated interpretations that led to pitfall putting at test security design principles.

8. Security meta reference architecture framework

EDITORIAL NOTE

the title of this section will need to be adjusted, this is only one small fraction of a meta reference architecture framework for security

This section defines a high-level framework that enumerates the concepts that can be utilised by a designer and/or architect in need to produce a security reference architecture, or a derived security solution architecture, or the derived actual security solution implementation and lifecycle.

8.1. Representation method

Each concept proposed in this Recommendation will be represented in the following uniform schema:
- ID: MUST
- Name: MUST
- Abbreviation: MAY
- Type: MUST
- Definition(s): MUST (at least one)
- Description: SHOULD
- Source(s): SHOULD
- Evolution: MAY
- Position in any security model: MAY
- Include: MAY
- Is included by: MAY
- Is obsoleted by: MAY
- Notes: MAY

Template table:

ID	
Name	
Abbreviation	
Type	
Definition(s)	
Description(s)	
Source(s)	
Evolution	

Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	

8.2. Security concerns

Whilst the identification of security concerns are an essential part of any design and architecture work, they are outside of the scope of this Recommendation.

8.3. Security architectural principles (SAP)

8.3.1. The system architecture is able to log and detect (SAP01)

Table — The system architecture is able to log and detect (SAP01)

ID	SAP01
Name	The system architecture is able to log and detect
Abbreviation	
Type	Security architectural principle
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	It is required that detection and logging capabilities are designed into the product/system security architecture. In this way, continuous learning and improvement could be supported. **EDITORIAL NOTE** This is a very good architecture consideration. Perhaps this should be split in detection and logging as 2 items. Is it a universal capability and should this stay in this Recommendation]

8.3.2. Ensure the system is scalable (SAP02)

Table — Ensure the system is scalable (SAP02)

ID	SAP02
Name	Ensure the system is scalable
Abbreviation	
Type	Security architectural principle
Definition(s)	
Description(s)	

Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	Scalability must be handled with great care. Scalability is a key property of security architecture. **EDITORIAL NOTE** This is an overall architecture characteristic. Shall it be defined here or elsewhere? In fact yes here as from Security perspective. It could be part of dependability, resiliency see [b-DEPENDABILITY]

8.3.3. Compartmentalize and de-couple whenever possible (SAP03)

Table — Compartmentalize and de-couple whenever possible (SAP03)

ID	SAP03
Name	Compartmentalize and de-couple whenever possible
Abbreviation	
Type	Security architectural principle
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	To segment or modularize the system design, is sound advice. Likewise, to reduce coupling to the minimum is a sensible goal. **EDITORIAL NOTE** That could be a security design principle, but what makes it different with defense in depth? It could be micro-segmentation, or network design, etc. In fact it is different from defence in depth because this is not layered

8.4. Security design principles (SDP)

8.4.1. Vulnerable components are unacceptable (SDP01)

Table — Vulnerable components are unacceptable (SDP01)

ID	SDP01
Name	Vulnerable components are unacceptable
Abbreviation	
Type	Security design principle
Definition(s)	
Description(s)	During security architecture design, it's required to either deprecate or refactor the vulnerable components of the product/system.
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	**EDITORIAL NOTE** This is (too) obvious, perhaps this should be re-interpreted from a different point of view on classification regarding an overall safety approach which would allow to classify this principle in the paradigm 'removal' in the 4 paradigms: prevention, tolerance, removal, forecasting.

8.4.2. Defense in depth (SDP02)

Table — Defense in depth (SDP02)

ID	SDP02
Name	Defense in depth
Abbreviation	N/A
Type	Security design principle
Definition(s)	DEF01) Information security strategy integrating people, technology, and operations capabilities to establish variable barriers across multiple layers and missions of the organization. DEF02) The application of multiple countermeasures in a layered or stepwise manner to achieve security objectives. The methodology involves layering heterogeneous security technologies in the common attack vectors to ensure the attacks missed by one technology are caught by another one.
Description(s)	Defense in depth is an approach in which a series of defensive mechanisms are layered in order to protect valuable data and information. This may be according to segmentation boundaries, etc. If one mechanism fails, the perpetrator must very soon face another security mechanism. This will make an attack more complex to conduct, and it will incur a greater cost in the attack. This will effectively make the attack less scalable and may even thwart the attack.
Source(s)	CNSSI 4009-2015 NIST SP 800-172 NIST SP 800-172A NIST SP 800-30 Rev1 under Defense-in-Depth from CNSSI 4009

	NIST SP 800-39 under Defense-in-Depth from CNSSI 4009 NIST SP 800-53 Rev.5 under defense in depth NISTIR 7622 under Defense-in-Depth NSTIR 8183 under Defense-in-depth from ISA/IEC 62443, ISO/IEC 62443 1-1 NSTIR 8183 Rev.1 under Defense-in-depth from ISA/IEC 62443 NSTIR 8183A Vol.2 under Defense-in-depth from ISO/IEC 62443 1-1 NSTIR 8183A Vol.3 under Defense-in-depth from ISO/IEC 62443 1-1
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	

8.4.3. Security coverage must be comprehensive and consistent (SDP03)

Table — Security coverage must be comprehensive and consistent (SDP03)

ID	SDP03
Name	Security coverage must be comprehensive and consistent
Abbreviation	
Type	Security design principle
Definition(s)	
Description(s)	Security features of the product/system typically comprise identification and authentication schemes, security protection for data in transit and data at rest, and security schemes for authorization and access protection. These functionalities need to be there and be as consistent and comprehensive as possible.
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	**EDITORIAL NOTE** this is indeed a security design principle but is it weak? The issue is that you can never be as complete as you want because of the 'capability/TCO/Risk appetite' curve

8.4.4. A threat modelling mindset must apply to security architecture design. (SDP04)

Table — A threat modelling mindset must apply to security architecture design. (SDP04)

ID	SDP04
Name	A threat modelling mindset must apply to security architecture design.
Abbreviation	

Type	Security design principle
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	Threat modelling is an activity normally associated with the design phase of a system, including security architecture design.

8.4.5. Zero Trust when considered a design security principle (SDP05)

Table — Zero Trust when considered a design security principle (SDP05)

ID	SDP05
Name	Zero Trust when considered a design security principle
Abbreviation	ZT
Type	Security design principle
Definition(s)	
Description(s)	Zero Trust is a security design principle and strategic approach that assumes no implicit trust is granted to assets or user accounts based solely on their physical or network location (i.e. , local area networks vs. the internet) or based on asset ownership (enterprise or personally owned). Instead, Zero Trust requires verifying the identity of anything and everything trying to connect to its systems before granting access, regardless of where the request originates. Under the Zero Trust model, security is not determined by the perimeter of the network but is instead based on strict identity verification, device health checks, least-privilege access, and micro-segmentation to minimize lateral movement within networks. Access to resources is granted on a need-to-know basis, and transactions are securely authenticated and authorized within a segmented environment.
Source(s)	DEF01 NIST SP 800-207 Zero trust provides a collection of concepts and ideas designed to minimize uncertainty in enforcing accurate, least privilege per-request access decisions in information systems and services in the face of a network viewed as compromised.
Evolution	2004 Jericho Forum introduces the principle of de-perimeterization. 2009 Google establishes Beyond Corp. 2010 Analyst John Kindervag introduces the "zero trust model" in a paper for Forrester Research. August 2020 NIST delivers NIST SP 800-207. Avril 2023 CISA delivers Zero Trust Maturity Model Version 2.0
Position in any security model	
Include	**EDITORIAL NOTE** Identify existing SDPs / Generate the missing SDPs from this list e.g. MFA, Encryption, Continuous Verification. Is SAP03 an SAP or an SDP? SDP14 (Continuous Verification)

	SDP07 (Principe of least-privilege): SDP11 (Micro-segmentation) SDP12 (MFASDP13 (Encrypt Data)
Is included by	
Is obsoleted by	
Notes	**EDITORIAL NOTE** This SDP has a number of issues – The first document from Forrester from 2010 that initiated this SDP is not available online anymore – The only formal definition is coming from NIST – There is a discrepancy between the definition in the NIST text and the NIST data base (Zero trust provides vs Zero trust is). As the text is the reference it breaks ITU-T Authors guide B.3.2 as there is no 'class' for the object. Then the next part of the clause if very broad "a collection of concepts and ideas" and then the indirect definition by objective "designed to" is followed by a non MECE list of elements – It is considered to propose an alternative definition on the form: "Zero trust is a security design principle which is composed of the following list of security design principles <list to be agreed on> that goal is to minimize …" where the <list to be agreed on> is a list of other SDPs in this Recommendation that are mutually exclusive though collectively exhaustive to form a MECE"

8.4.6. Minimize the attack surface area (SDP06)

Table — Minimize the attack surface area (SDP06)

ID	SDP06
Name	Minimize the attack surface area
Abbreviation	
Type	Security design principle
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	

8.4.7. The principle of least privilege (SDP07)

Table — The principle of least privilege (SDP07)

ID	SDP07
Name	The principle of least privilege
Abbreviation	PoLP
Type	Security design principle

Definition(s)	DEF01 NIST 800-53 R5 AC-6 Control Statement: Employ the principle of least priviledge, allowing only authorized accesses for users (or processes acting on behalf of users) that are necessary to accomplish assigned organizational tasks.
Description(s)	**EDITORIAL NOTE** Users and devices are given the minimum access necessary to perform their duties, reducing the potential impact of a breach. It refers to the practice of limiting access rights for users (and systems) to the bare minimum necessary to perform their functions. This means that a user, program, or process should have only the privileges which are essential for its intended function, nothing more. Implementing the least privilege principle helps to reduce the attack surface of a system by limiting access to critical systems and data to only those entities that require it to perform their duties. This can significantly mitigate the potential damage from various security threats, such as malware infections or the actions of malicious actors. By ensuring that users and systems operate using the minimal set of privileges, organizations can better protect sensitive information and critical infrastructure from unauthorized access and exploitation. The principle of least privilege can be applied across various aspects of IT environments, including user permissions, software execution, system processes, and network access. It is often enforced through user account management processes, role-based access control (RBAC), access control lists (ACLs), and other security mechanisms designed to control access and privileges effectively.
Source(s)	**EDITORIAL NOTE** To research into: ISO/IEC 27001 and in particular ISO/IEC 27001:2013 A.9 NIS Special Publication 80-53 ISO/IEC 15408 The Common Criteria for Information Technology Security Evaluation (CC) Payment Card Industry Data Security Standard (PCI/DSS) The Center for Internet Security (CIS) Controls Federal Information Processing Standards (FIPS)
Evolution	The principle of least privilege (PoLP) is widely attributed to Jerome Saltzer and Michael D. Schroeder, who first articulated it in their seminal paper titled "The Protection of Information in Computer Systems," published in 1975 as part of the Proceedings of the IEEE, Vol. 63, No. 9. This paper laid out a set of design principles for securing information in computer systems, among which the principle of least privilege played a crucial role. Saltzer and Schroeder were part of the research community at MIT's Project MAC, which later became the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL). Their work was foundational in the field of computer security, influencing not only academic research but also the practical design and implementation of secure computing systems. The principle of least privilege is one of several key principles they introduced, which also include concepts like economy of mechanism, fail-safe defaults, and separation of privilege. These principles have since become standard guidelines in the design and operation of secure systems. While the formal articulation of the principle dates back to Saltzer and Schroeder's 1975 paper, the underlying concept of minimizing access or privilege to what is necessary for a particular purpose has been a common practice in security-sensitive environments even before its formalization in the context of computer security.
Position in any security model	
Include	
Is included by	SDP05 (ZT)
Is obsoleted by	

Notes	

8.4.8. Separation of duties (SDP08)

Table — Separation of duties (SDP08)

ID	SDP08
Name	Separation of duties
Abbreviation	
Type	Security design principle
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	**EDITORIAL NOTE** Need to clarify relationship with Least privilege

8.4.9. Security by Design is the most cost-effective approach to security (SDP09)

Table — Security by Design is the most cost-effective approach to security (SDP09)

ID	SDP09
Name	Security by Design is the most cost-effective approach to security
Abbreviation	
Type	Security design principle
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	Security is vital for all critical infrastructures and should be designed into systems and operations from the beginning, rather than being applied after the systems have been implemented. **EDITORIAL NOTE** There is another definition earlier:

	Security by design is an approach in development that helps to focus on making a system as secure as possible already in the development process. It also helps to focus on best design practices. Where is it defined in a normative term It is not always feasible It is not the solution because of judge and party see CG-SECAPA discussion What's about security by implementation, migraton, etc.

8.4.10. Never trust, always verify (SDP10)

Table — Never trust, always verify (SDP10)

ID	SDP10
Name	Never trust, always verify
Abbreviation	
Type	Security Design Principle
Definition(s)	The premise that trust is never granted implicitly but must be continually evaluated.
Description(s)	A restatement of the Zero Trust premise.
Source(s)	NIST SP 800-207
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	

8.4.11. Micro-segmentation (SDP11)

Table — Micro-segmentation (SDP11)

ID	SDP11
Name	Micro-segmentation
Abbreviation	
Type	Security Design Principle
Definition(s)	DEF01 NIST SP 800-215: Microsegmentation is a security design practice where an internal network (e.g., in the data center, cloud provider region) is divided into isolated segments so that the traffic in and out of each segment can be monitored and controlled.
Description(s)	Networks are divided into small, secure zones to maintain separate access for separate parts of the network. This limits an attacker's ability to move laterally across a network. The primary purpose of microsegmentation is to provide a degree of isolation to prevent attack escalation.
Source(s)	NIST SP 800-215, 5.1
Evolution	
Position in any security model	
Include	
Is included by	

Is obsoleted by	
Notes	

8.4.12. Multi-Factor Authentication (SDP12)

Table — Multi-Factor Authentication (SDP12)

ID	SDP12
Name	Multi-Factor Authentication
Abbreviation	MFA
Type	Security Design Principle
Definition(s)	DEF01 NIST SP 800-63 Revision 3 Appendix A: Multi-Factor Authentication (MFA): An authentication system that requires more than one distinct authentication factor for successful authentication. Multi-factor authentication can be performed using a multi-factor authenticator or by a combination of authenticators that provide different factors. The three authentication factors are something you know, something you have, and something you are.
Description(s)	The use of multiple verification methods to ensure that a user or device is granted access only after successfully presenting two or more pieces of evidence to an authentication mechanism.
Source(s)	NIST SP 800-63 Revision 3 Appendix A
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	

8.4.13. Encrypt data (SDP13)

Table — Encrypt data (SDP13)

ID	SDP13
Name	Encrypt data
Abbreviation	
Type	Security Design Principle
Definition(s)	
Description(s)	Encrypting data at rest and in transit to protect the integrity and confidentiality of the data, even if a network is compromised.
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	

8.4.14. Continuous verification (SDP14)

Table — Continuous verification (SDP14)

ID	SDP14
Name	Continuous verification
Abbreviation	
Type	Security Design Principle
Definition(s)	
Description(s)	Trust is never assumed and must be continually reassessed. Authentication and authorization are required for all users and devices seeking access to resources, regardless of their location.
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	

8.4.15. De-perimeterization (SDP15)

Table — De-perimeterization (SDP15)

ID	SDP15
Name	De-perimeterization
Abbreviation	
Type	Security Design Principle
Definition(s)	
Description(s)	
Source(s)	Jerico Forum
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	

8.5. Security design considerations (SDC)

8.5.1. Robustness is a prerequisite for a security architecture (SDC01)

Table — Robustness is a prerequisite for a security architecture (SDC01)

ID	SDC01
Name	Robustness is a prerequisite for a security architecture

Abbreviation	
Type	Security design consideration
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	In a dynamic system, a state of robustness is not inherently stable and cannot be expected to last forever. However, it is still a necessary requirement that all components be robust. If a component of the product/system is weak, it's required to remedy the situation. **EDITORIAL NOTE** Robustness is it a synonym for resiliency?

8.5.2. Threat landscape awareness is a prerequisite (SDC02)

Table — Threat landscape awareness is a prerequisite (SDC02)

ID	SDC02
Name	Threat landscape awareness is a prerequisite
Abbreviation	
Type	Security design consideration
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	In order to take appropriate countermeasure, it is important to know what kind of threats the system is likely to face. Therefore, it is required to conduct threat landscape investigations, which is a continual process.

8.5.3. Awareness of the Cyber Kill Chain is necessary (SDC03)

Table — Awareness of the Cyber Kill Chain is necessary (SDC03)

ID	SDC03
Name	Awareness of the Cyber Kill Chain is necessary
Abbreviation	
Type	Security design consideration

Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	Advanced persistent threat (APT) actors will tend to follow a certain set of steps to attack a system. These steps are called the "kill chain". Kill chain knowledge is no silver bullet but kill chain awareness is nevertheless very important. **EDITORIAL NOTE** a very important pre-requisite and best practice

8.5.4. Fallback and backwards compatibility must be managed (SDC04)

Table — Fallback and backwards compatibility must be managed (SDC04)

ID	SDC04
Name	Fallback and backwards compatibility must be managed
Abbreviation	
Type	Security design consideration
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	In a dynamic system, it is required to manage fallback and backwards compatibility during the security architecture design. **EDITORIAL NOTE** In practice even downgrading a workstation for whatever reason is close to impossible today! It is ideal but not always practical for many reasons including business reasons

8.5.5. Single point of failure must be avoided (and planned for) (SDC05)

Table — Single point of failure must be avoided (and planned for) (SDC05)

ID	SDC05
Name	Single point of failure must be avoided (and planned for)
Abbreviation	

Type	Security design consideration
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	With serious consideration of the drawbacks, a single point of failure could be avoided. **EDITORIAL NOTE** This is part of Resiliency and redundancy architecture characteristic

8.5.6. All security functions [must][should] be upgradable, replaceable and updatable (SDC06)

Table — All security functions [must][should] be upgradable, replaceable and updatable (SDC06)

ID	SDC06
Name	All security functions [must][should] be upgradable, replaceable and updatable
Abbreviation	
Type	Security design consideration
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	All security functions will need to be upgradable and replaceable, which can pose a lot of challenges for security functionality. **EDITORIAL NOTE** What means replaceable? Vendor lock-in. Is it a security function or a solution. A security function is not upgradable by semantic]

8.5.7. There must be strong detection and response capabilities (SDC07)

Table — There must be strong detection and response capabilities (SDC07)

ID	SDC07

Name	There must be strong detection and response capabilities
Abbreviation	
Type	Security design consideration
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	Reactive security measures are for the dynamic cases, and often for unexpected events. So, detection and response capabilities, including the full gauntlet of recovery and incident investigations, are necessary to be supported.

8.5.8. Plan for success and a long-term future (SDC08)

Table — Plan for success and a long-term future (SDC08)

ID	SDC08
Name	Plan for success and a long-term future
Abbreviation	
Type	Security design consideration
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	The passage of time almost directly translates into change. And, invariably, any successful large-scale systems will be long lived. This literally translates into requirements to embrace change.

8.6. Security design best practices (SDB)

8.6.1. Failures provide invaluable information (SDB01)

Table — Failures provide invaluable information (SDB01)

ID	SDB01
Name	Failures provide invaluable information
Abbreviation	
Type	Security design best practice
Definition(s)	
Description(s)	

Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	To learn from failure is essential. Failures provide vital information about how the system actually works. To learn from security failures in other systems is also important.

8.6.2. System interfaces and exposure should be explicitly defined (SDB02)

Table — System interfaces and exposure should be explicitly defined (SDB02)

ID	SDB02
Name	System interfaces and exposure should be explicitly defined
Abbreviation	
Type	Security design best practice
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	Be explicit about intended exposure. To be explicit about intended exposure does not guarantee that the attack surface is well-contained, but it will at least indicate that the problem has been considered

8.6.3. Be explicit. Do not assume (SDB03)

Table — Be explicit. Do not assume (SDB03)

ID	SDB03
Name	Be explicit. Do not assume
Abbreviation	
Type	Security design best practice
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	

Notes	A sound design is normally also a more secure design. A clean and transparent design will contribute towards this goal. However, it is under most conditions also an unattainable goal. Still, this is not the kind of goal that one expects to reach, it more a guideline for direction. One factor that contributes quite a lot is explicitness. Do not assume anything. If it is indeed important, then state it explicitly. This is sound advice for system designs at large, and even more so for security designs.

8.6.4. Known vulnerabilities should be prioritised and fixed accordingly, through different security and protection levels. (SDB04)

Table — Known vulnerabilities should be prioritised and fixed accordingly, through different security and protection levels. (SDB04)

ID	SDB04
Name	Known vulnerabilities should be prioritised and fixed accordingly, through different security and protection levels.
Abbreviation	
Type	Security design best practice
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	An attacker would need to exploit some kind of vulnerability in order to successfully carry out an attack. This is not to suggest that every vulnerability is equally important or need urgent attention. It simply means that all known vulnerabilities could be fixed through different security and protection levels. Sometimes it may suffice to reduce the exposure to provide an effective stopgap mitigation **EDITORIAL NOTE** As there might be thousands of vulnerabilities, prioritisation is essential and the most severe/critical ones should be addressed.

8.6.5. Fail securely (SDB05)

Table — Fail securely (SDB05)

ID	SDB05
Name	Fail securely
Abbreviation	
Type	Security design best practice
Definition(s)	
Description(s)	
Source(s)	

Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	

8.6.6. Avoid security by obscurity (SDB06)

Table — Avoid security by obscurity (SDB06)

ID	SDB06
Name	Avoid security by obscurity
Abbreviation	
Type	Security design best practice
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	

8.6.7. Keep security simple (SDB07)

Table — Keep security simple (SDB07)

ID	SDB07
Name	Keep security simple
Abbreviation	
Type	Security design best practice
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	

8.6.8. Asset clarification (SDB08)

Table — Asset clarification (SDB08)

ID	SDB08
Name	Asset clarification
Abbreviation	
Type	Security design best practice
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	**EDITORIAL NOTE** change the title this is about Asset identification and classification

8.6.9. Establish secure defaults (SDB09)

Table — Establish secure defaults (SDB09)

ID	SDB09
Name	Establish secure defaults
Abbreviation	
Type	Security design best practice
Definition(s)	
Description(s)	
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	**EDITORIAL NOTE** massive difference between setting defaults in Vendor X products vs Vendor Y products

8.7. Security design constraint (SDX)

8.7.1. Juvenal (SDX01)

Table — Juvenal (SDX01)

ID	SDX01
Name	Juvenal
Abbreviation	

Type	Security design constraint
Definition(s)	
Description(s)	Despite the fact that security of some elements in the system can be proved, there is no definite way to measure and compare security of the whole system. This will be called the Juvenal security design constraint in reference to the famous quote: 'sed quis custodiet, ipsos custodet' which can be interpreted as 'who guards the guards'. This security design constraint represents a key 'glass roof' that may be pushed, may be deformed but doesn't seem to have any possibility to be pierced.
Source(s)	
Evolution	
Position in any security model	
Include	
Is included by	
Is obsoleted by	
Notes	

9. Evolutionary considerations

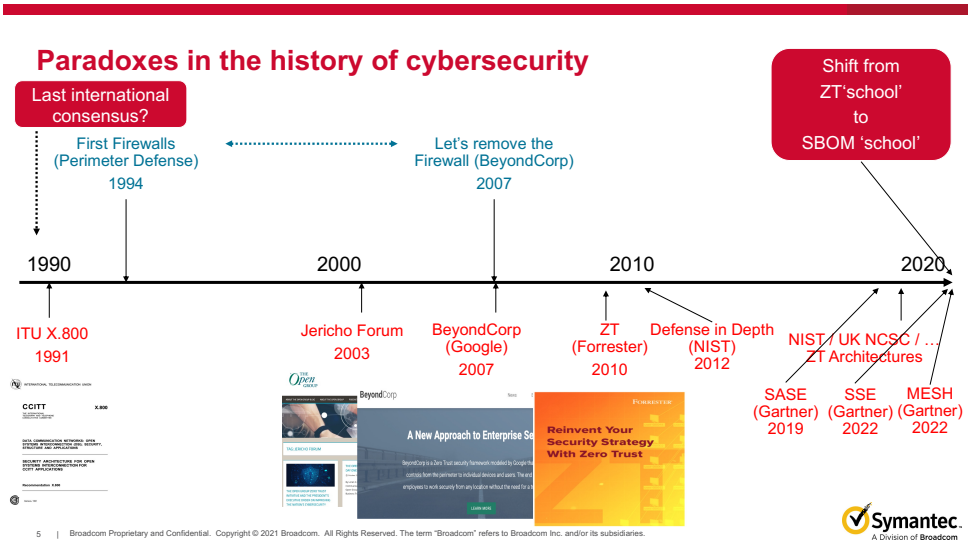

Figure 1

10. Security design principles relationships

This section studies the relationships between security design principles with the objective to:
– identify overlaps,
– identify those who are part of a MECE
– build Euler-Venn diagrams.

10.1. Not a security design principle

There are a number of concepts in the industry that depending on the context are not security design principles.

Examples:
– SASE
– SSE
– MESH

11. Consideration on Designer and Architect Roles

EDITORIAL NOTE

This section is important but will require a lot of rewording

Designers and architects form a key constituency of this Recommendation.
– They play an important if not existential role in the success of a solution.
– They can too be the limit to this success.

11.1. Context for the role

11.1.1. Designer and architect jobs across domains

In military engineering and way later in civil engineering (as this is a domain where humans have a long experience) designers and architects have rather well codified job descriptions with full curricula that are not only licensed but deliver diploma which not only gives the right to the architect to do his job, but comes too with responsibilities and liabilities.

If a bridge falls down, both from a legal and an insurance perspective, the process will inevitably lead to the question of whether or not the architect is responsible (liable) or not. His/her responsibility may be engaged.

11.1.2. Designer and architect jobs in ICT

The ICT industry incrementally recognized the problem and attributed its solution to architects and designers which were allocated in various types and companies functions in IT and ICT we observe many differences, at this stage:
– IT and ICT are domains that are much younger by orders of magnitude than civil engineering
– The role/job of an architect is extremely recent and was mostly hidden in the wordings 'software engineer', etc.
– The role/job and covers many subtypes:
  • Software architect
  • System architect
  • Solution architect
  • Etc.
– For a long time there were no codification and even trainings or certification for this job until TOGAF arrived in 1995 and yet, even today, like anything, it has limits
– There are no liabilities attached to any architect. An architect making a mistake at design level has absolutely no risk even if (lived stories) it could incur enormous costs and liabilities for the 'customer' and for the 'provider' of the architecture.

11.1.3. Different types of ICT Designer and Architects

Firstly, at product and service level, there are various types of architects (the list is not meant to be exhaustive)

Table 1 — Architect and Designer types

Architect type	Coverage	Organization
Software Architect	Covers the architecture of a software that needs to be developed	Engineering / R&D

Architect type	Coverage	Organization
Product Architect	Covers the end to end architecture of product that needs to be developed	Engineering / R&D
Designer	Covers the full design of a product or groups of products including other design criteria such as Societal, Technical, Ecological, Environmental, Political, Human Factor, etc.	Engineering / R&D
Security Architect	Covers the security aspect of a solution and proposes either a security architecture or a security by design 'design'	SoC / CISO / etc.
System Architect	Covers the design of an entire set of systems (hardware, software, etc.) that needs to be put in production for a given period of time (usually years or more)	Field
Solution Architect	Covers the end to end solution (hardware, software, professional services, partners, compliancy, etc.) for a given customer	Field
Technical Directors	Office of the CTO and CTOs do have a view on design in terms of internal standardisation, design directions, harmonisation, composability and do participate in the organization transformation, breaking the silos or contributing to the collaboration and coordination between the silos based on design approaches	Office of CTO

11.1.4. The nature of the job

The architects and designers have a pivot job in each organization because they have to produce deliverables that will take into

EDITORIAL NOTE

the first and second bullets should be checked vs above terminology, e.g. shouldn't we use the term 'concern' in the first bullet

– In one hand, the considerations of definitions, standards, requirements, limits and constraints
– On the other hand, the whole lifecycle of a product or a service

The diagram below shows this pivot role on the top and on the bottom shows a number of dimensions that make the characteristics of the architect in his core and deepest nature.

EDITORIAL NOTE

Harmonise the cycle with the cycle proposed in section 9.4

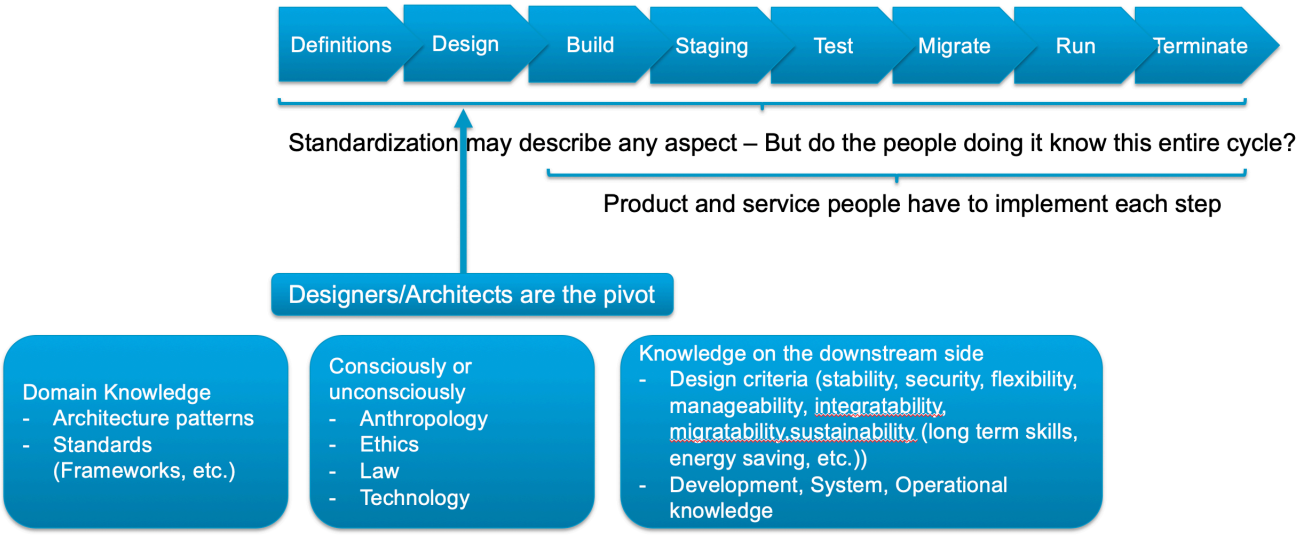

Figure 2 — The architect and designers play a pivot role

As well it is important that each architect / designer, will come with his own approach which will likely be unique in itself. The architect / designer, could consider his/her deliverables vs the four below dimensions:

– Anthropology
– Ethics
– Law
– Technology

In special conditions though, especially when no human being had any previous experience, one needs to consider the reverse order:

– Technology
– Law
– Ethics
– Anthropology

12. Examples

EDITORIAL NOTE

this section will need a lot of curation but will be done after section 7, 8 and 9 are complete

12.1. Key Concepts for Cyber Security

12.1.1. Concept #1

Resilience should be the overall strategy for ensuring business continuity: When focusing on resilience in general, organizations must consider safety, security, and reliability of the processes and the delivery of their services. Resilience includes security measures that can mitigate impacts, not only before incidents (identify & prevent), but also during such incidents (detect & respond) and after incidents have been resolved (recover).

EDITORIAL NOTE

is this a security design principle? Or is it a context where security design principles should apply? Or is it a context that imposes new security design principles, or constraints or

composition issues? This infers a new section on composition/usage, even perhaps AFTER Kishor's section

12.1.2. Concept #3

IT and OT are similar but different: Technologies in Operational environments (called OT) have many differing security constraints and requirements from Informational Technologies (IT) environments.

EDITORIAL NOTE

- Same as concept #1 this is not a design principle but areas of applicability
- We should look at transformation of NT, AI, IOT, OT, IT

12.1.3. Concept #4

Risk assessment, risk mitigation, and continuous update of processes are fundamental to improving security: Based on an organization's business requirements, its security risk exposure must be determined (human safety, physical, functional, environmental, financial, societal, reputational) for all its business processes.

12.1.4. Concept #5

Cyber security standards and best practice guidelines for OT environments should be used to support the risk management process and establish security programs and policies: at the right time.

Appendix I

A comprehensive and granular Cyber Security Architecture imperative for Civic Infrastructure.

(This appendix does not form an integral part of this Recommendation.)

I.1. Context

International law defines Four Global Commons (natural assets outside national jurisdiction) which are the earth's natural resources i.e. the High Seas, the Atmosphere, Antarctica, and Outer Space. Cyberspace is the 5th Global Common. It is also considered as the 5th Dimension beyond the 3 dimensions of Space & 4th dimension being the Time.

Challenges that all economies are facing today in safeguarding the security and privacy of its ecosystem including citizen are - Transnational Nature of Cyber Crime, 'Cultural' Vulnerabilities, Internet Resilience and Threat Landscape.

Cyber risk threat vectors have evolved rapidly, and attacks have become increasingly sophisticated, deliberate, and unrelenting in nature. In the digital era, trust is a complex issue fraught with myriad existential threats to the enterprise. And while disruptive technologies are often viewed as vehicles for exponential growth, tech alone can't build long-term trust. Every aspect of an organization disrupted by technology represents an opportunity to gain or lose stakeholders' trust. Leaders are approaching trust not as a compliance issue but as a business-critical goal. For this reason, leading organizations are taking a 360-degree approach to maintain the high level of trust their stakeholders expect.

The new paradigm of Smart Grid, Smart Home, Smart Building, Smart Manufacturing, Smart City already complicated by the 'Internet of Things' & Internet of 'Everything' made further complex by the 5G, Artificial Intelligence, Machine Learning, Blockchain & Quantum Computing, make it truly complex to develop and embed comprehensive Security, Privacy and Trustworthiness attributes in the products, systems and solutions for any use case or application - be it consumer, commercial, industrial, automotive or strategic domains like civic infrastructure.

The recent evolution of disruptive technologies and digitalization compounded by the Covid 19, changing geopolitical situations, and increasing cyber-attacks bring a whole new set of challenges for the Security and Security Evaluation Methodologies for complex nature & architectures of Civic Infrastructures of the nation leveraging the IT & Communication Networks evolving to meet these rising needs of the Society.

The highly protected Networks for the 'Civic Infrastructures' need to give access to the consumers for Consumer Engagement and Participation in these Smart (Digital) Infrastructures to meet the true drivers of setting them up. These large Smart Networks are actually highly complex 'Systems of Systems' and "Networks of Networks', and thus create fresh challenges in the Security Paradigm and development of Protection Profiles.

It is evident that Cyber Security is a very complex paradigm, and with evolving new technologies, requirements, and ever-increasing Attack Surface the vulnerabilities are rising many folds with time. In such a dynamic scenario, it is required to develop a Cyber Security Strategy to make our Critical Infrastructure comprehensively Safe, Secure, Resilient and Trustworthy.

I.2. Imperative

The civic infrastructure cyberthreat landscape is rapidly evolving and expanding, with more frequent attacks, more numerous and varied threat actors, and increasingly sophisticated malware and tools that are more widely available and sometimes indiscriminately deployed. Civic infrastructure operations

are among the most frequently attacked targets, increasingly by nation-state actors aiming for disruption and even destruction through ICS.

It would be reasonable to assume that all the stakeholders have already understood the urgency of ensuring the Security & Resilience of Civic Infrastructure; however, the initiatives and approaches already adopted and/or being adopted by the different arms of the governments are quite arbitrary and random, considering point solutions with limited effectiveness to mitigate highly complex cyber threats.

Improving cyber safety and resilience requires all stakeholders to act together at scale and in a coordinated way, including governments, the engineering professionals, operators of civic infrastructure and other systems, and developers of products and components. The evolving nature of the challenges will require continual responsiveness and agility by governments and other stakeholders.

The need for proven, scalable, and standards-based solutions for Civic Infrastructures' deployment scenario, with inherent complexity and trade-offs, requires specialized, skilled, and multi-stakeholder engagement. THUS, it is required to undertake this task of global importance, which shall make a significant contribution in building a "Robust Foundation for Civic Information Infrastructures" along with paving ways to make our community "secure & sustainable".

The only approach would be to adopt top-down approach to standardization starting at the system or system-architecture rather than at the product level. It is required to Study & Analyse the diverse Use Cases, Applications and corresponding Stakeholders & their respective requirements to understand their respective Characteristics and concerns. Develop a Granular Civic Infrastructures' Cyber Security Architecture mapping all the security, privacy, safety, resilience characteristics with the Granular Civic Infrastructure Architecture.

Based on the developed Cyber Security Reference Architecture, the diverse standards shall need to be mapped to well identified Stakeholders' concerns and diverse Products, Systems & Solutions being deployed. In accordance with the appropriate Standards identified & mapped, a comprehensive Compliance Testing Framework followed by granular Testing Schemas shall need to be developed based on which the Testing Infrastructure could be created.

Unless, the aforementioned milestones are achieved, the Security Compliance & Testing Strategy shall NOT deliver the desired results.

I.3. Conclusion

Innovation and technology development are accelerating. Strategic plans and roadmaps are needed to help ensure that the market is suitably served with best practices that is pertinent to the goals and context of this very large market.

The multiplicity of technologies and their convergence in many new and emerging markets, however, particularly those involving large-scale infrastructure demand a top-down approach to standardization starting at the system or system-architecture rather than at the product level. Therefore, the systemic approach in standardization work can define and strengthen the systems approach throughout the technical community to ensure that highly complex market sectors can be properly addressed and supported. It promotes an increased co-operation with many other standards-developing organizations and relevant non-standards bodies needed on an international level.

Given the scale, moving forward through the labyrinth of Disruptive Technologies cannot be successfully, efficiently, and swiftly accomplished without standards. The role of standards to help steer and shape this journey is vital. Standards provide a foundation to support innovation. The Standards support our need to balance agility, openness, and security in a fast-moving environment. The Standards provide us with a reliable platform to innovate, differentiate and scale up our technology

development. They help to control essential security and integrate the right level of interoperability. Standards help ensure cyber security in ICT and IoT systems (Digital & Cyber Physical systems). Standards capture best practices and set regulatory compliance requirements, which is crucial for the sustainable Digital Transformation of the Critical Infrastructure.

It is imperative to delve into the security, privacy & trustworthiness aspects, and implications of the new paradigm of "Digital Infrastructure" and "Internet of Things" that the pervasive computing has enabled, thus raising new challenges for the 'IT & Communication Security' Development & Evaluation Eco-system. Hence, needing a new rigorous and vigorous effort in developing a "Comprehensive Cyber Security, Resilience & Trustworthiness" Strategy Framework encompassing all the critical domains and Stakeholders' classifications and their respective imperatives from Cyber Security & Resilience & Trustworthiness Perspective.

Bibliography

[ISO/IEC/IEEE 42010:2022] ISO/IEC/IEEE 42010:2022 (2022), *Software, systems and enterprise—Architecture description*, Second edition.

[b-RFC9413] IETF RFC 9413 (2023), *Maintaining Robust Protocols*.

[b-DEPENDABILITY] AVIZIENIS, A., J.-c. LAPRIE, B. RANDELL and C. LANDWEHR. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*. vol. 1 no. 1, pp. 11–33. Institute of Electrical and Electronics Engineers (IEEE). 2004. DOI: DOI 10.1109/tdsc.2004.2. ISSN: issn.print 1545-5971. http://ieeexplore.ieee.org/document/1335465/.

[b-NOD] *The Nature of design*.

[b-SecArch-Design] KØIEN, Geir M. A Philosophy of Security Architecture Design. *Wireless Personal Communications*. vol. 113 no. 3, pp. 1615–1639. Springer Science and Business Media LLC. 2020. DOI: DOI 10.1007/s11277-020-07310-5. ISSN: issn.print 0929-6212. ISSN: issn.electronic 1572-834X. https://link.springer.com/10.1007/s11277-020-07310-5.

[b-BCE Updated] b-BCE Updated (2023), *Transforming Remote Access with Google BeyondCorp Enterprise.*.

[b-BCE Usenix] b-BCE Usenix (2014), WARD, R. and B. BEYER. *BeyondCorp A New Approach to Enterprise Security*. In. (login.) n.p.: Usenix. 2014. vol. 39 no. 6. b-BCE Usenix. BeyondCorp A New Approach to Enterprise Security.