NATO ARCHITECTURE FRAMEWORK

Version 4

Architecture Capability Team
Consultation, Command & Control Board
January 2018

Acknowledgments for NAFv4 Publication

Throughout the development of version 4 of this publication numerous individual experts of NATO Nations participated, resulting in this significant achievement:

The realization of the NATO Architecture Framework.

This work would not have been possible without the continuous support of the Ministries of Defence of United Kingdom and France, and the NATO Science and Technology Organization.

Also special thanks goes to Partner Nations and Industry Partners for their unwavering support in assigning and providing their best professional resources in the architecture domain.

The NATO Architecture Framework is a substantial achievement for the Architecture Capability Team under the Consultation, Command and Control Board. Each member of the Architecture Capability Team worked determinedly over the last four years to provide extensive professional guidance and personal effort in the development of this product.

The Architecture Capability Team is grateful to all for their contributions to this effort.

C3B Architecture CaT
Providing Insight And Commonality

CONTENTS

Chapter 1 - Introduction

Chapter 2 - Methodology

Chapter 3 - Viewpoints

TABLE OF FIGURES

Chapter 1 - Introduction

Chapter 2 - Methodology

Chapter 3 - Viewpoints

TABLE OF TABLES

Chapter 2 - Methodology

Chapter 3 - Viewpoints

Chapter 1 - Introduction

1 GENERAL

1.1 Purpose

1.1.1 Architecting is a practice for conducting enterprise analysis, design, planning, and implementation, using a holistic engineering approach at all times, for the implementation of strategy.

1.1.2 Architecting applies principles and practices to guide organizations through the business/mission, information, application and technology changes necessary to implement their strategies[1].

1.1.3 Good architecture practices include the usage of architectural artefacts to describe, assess, evaluate and document relevant aspects of an architecture.

1.1.4 The NATO Architecture Framework (NAF) provides a standardized way to develop architecture artefacts, by defining:

- Methodology – how to develop architectures and run an architecture project (Chapter 2),
- Viewpoints – conventions for the construction, interpretation and use of architecture views for communicating the enterprise architecture to different stakeholders (Chapter 3),
- Meta-Model – the application of commercial meta-models identified as compliant with NATO policy (Chapter 4), and
- a Glossary, References and Bibliography (Chapter 5).

1.2 Aim

1.2.1 The aim of the NATO Architecture Framework Version 4 (NAFv4) is to provide a standard for developing and describing architectures for both military and business use.

1.3 Objectives

1.3.1 The objectives of the framework are to:

- provide a way to organize and present architectures to stakeholders,
- specify the guidance, rules, and product descriptions for developing and presenting architecture information,
- ensure a common approach for understanding, comparing, and integrating architectures,
- act as a key enabler for acquiring and fielding cost-effective and interoperable capabilities, and
- align with architecture references produced by international standard bodies (International Standards Organization (ISO), Institute of Electrical and Electronic Engineers (IEEE), The Open Group (TOG), Object Management Group (OMG) etc).

1.4 Scope of NAF Documentation

1.4.1 This document provides an overview of the architecture concepts, the structure and the framework, and indicates where to find more specific information. It also describes, in general terms, the typical content and format of NAFviewpoints, and the relationship with the commercial meta-model constructs.

1.5 Reason for Change

1.5.1 NAFversion 3 (NAFv3) was issued in 2007[2] to support alliance interoperability through the coherent use of architectures, and provide for the re-use of architecture artefacts and products to facilitate the description of systems and applications. However, NAFv3:

- was not consistently applied by projects,
- did not provide a common architecture approach,

---

1 A Common Perspective on Enterprise Architecture, The Federation of Enterprise Architecture Professional Organizations.

2 NAFv3 was issued as Annex 1 to AC/322-D(2007)0048, was released to the public with AC/322-D(2015)0009 and replaced MODAF Version 1.2.004.

- became challenging to maintain due to limited technical resources, and
- did not align with major terms and concepts in the following international standards:
  - ISO/IEC/IEEE 42010 Systems and Software Engineering – Architecture Description,
  - ISO/IEC CD 42020 Systems and Software Engineering – Architecture Processes,
  - ISO/IEC 42030 Systems and Software Engineering – Architecture Evaluation,
  - The Open Group Architecture Framework (TOGAF) Version 9.1,
  - ISO/IEC/IEEE 15288 Systems and Software Engineering – System Lifecycle Processes,
  - ISO 15704 Industrial automation systems – Requirements for enterprise-reference architectures and methodologies.

1.5.2 NAFv4 addresses the above limitations and is a step towards a single Architecture Framework across NATO and Nations.

2 WHAT IS ARCHITECTURE?

2.1 Description

2.1.1 ISO/IEC/IEEE 42010 describes architecture as:

“The fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution”.

2.1.2 In the case of the NAF, a system is anything that can be considered with a systemic approach, such as a:

- product,
- service,
- information system,
- system of systems, or
- enterprise.

2.1.3 However, a description of architecture can be started before any identification of systems. This is the case when the description starts with a pure operational description or a set of operational capabilities explaining what the user needs.

2.2 Why Develop Architectures?

2.2.1 Architectures are developed for many purposes and their development can be described as both a process and a discipline. Architectures aid the development of systems that deliver solutions that can meet an organization’s needs in order to achieve its mission.

2.2.2 Examples of why architecture is required include:

- planning the transition of capability throughout its lifecycle,
- achieving greater flexibility, adaptability and capacity for cost effective acquisitions and building Multi-national systems for supporting operations,
- understanding and mitigating risks,
- better adaption to changes in the business landscape, industry trends and regulatory environment,
- aligning business and technology to the same set of priorities,
- planning, and managing, investment and controlling expenditure to business, and
- improving communication within technical domains and between Communities of Interest (CoI).

3 WHAT IS AN ENTERPRISE ARCHITECTURE?

3.1 Description

3.1.1 An Enterprise Architecture (EA) is a way of formalizing stakeholder concerns and presenting them in the context of the enterprise. For example EA can encompass both business and technical concepts to emphasize the dependencies between them. This approach enables change to proceed with a clearer understanding of the touch-points and problem areas. EA takes a holistic approach in order to manage problems associated with the system-of-interest to show the interaction of technology and business processes.

3.1.2 The purpose of EA is to optimize across the enterprise, the often fragmented legacy of processes (both manual and automated) and systems, into an integrated environment that is responsive to change and supports the delivery of the business strategy. The purpose of EA is not to model the entire enterprise.

3.1.3 An EA should encompass the architecture definition process as described by ISO/IEC/IEEE 15288-2015.

"The purpose of the Architecture Definition process is to generate system architecture alternatives, to select one or more alternative(s) that frame stakeholder concerns and meet system requirements, and to express this in a set of consistent views.

Iteration of the Architecture Definition process with the Business or Mission Analysis process, System Requirements Definition process, Design Definition process, and Stakeholder Needs and Requirements Definition process is often employed so that there is a negotiated understanding of the problem to be solved and a satisfactory solution is identified. The results of the Architecture Definition process are widely used across the life cycle processes. Architecture definition may be applied at many levels of abstraction, highlighting the relevant detail that is necessary for the decisions at that level."

---

4 WHAT IS AN ARCHITECTURE FRAMEWORK?

4.1 Description

4.1.1 An architecture framework is a specification of how to organize and present an enterprise through architecture descriptions. ISO/IEC/IEEE 42010 describes an architecture framework as:

"The conventions, principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders".

4.1.2 An evolution of this reference proposes the following definition:

"The conventions, principles and practices for the architecture activities established within a specific domain of application and/or community of stakeholders".

4.1.3 It consists of a set of standard viewpoints which ISO/IEC/IEEE 42010 describes as:

"The work product establishing the conventions for the construction, interpretation and use of architecture views to frame specific system concerns".

4.1.4 To manage complexity, NAFv4 has been developed and defines a standard set of viewpoints which each have a specific purpose. NAF define viewpoints in terms of the concerns they address.

5 THE STRUCTURE OF THE NATO ARCHITECTURE FRAMEWORK (NAF)

5.1 Introduction

5.1.1 The NAF is designed to ensure that architectures developed adhering to it can be understood, compared[3], justified and related across many organizations, including NATO and other National Defence initiatives.

5.1.2 The traditional approach to development has often resulted in a collection of disparate systems procured and provided by the Nations that may be interconnected but were never interoperable such that the combination was aligned with an organization's goal.

5.1.3 As a result of this situation, systems failed to bring the expected benefits like interoperability, speed of operation, cost reduction and flexibility to change.

5.1.4 The solution to this is to think strategically and understand an organization's overall objectives. From these objectives the actual content and the structure of the systems can be derived. The rules, constraints and guidelines on how to develop capabilities and systems including information systems to support the business, is a central element for architects.

5.1.5 Architectures must transform strategy into the content of manageable and executable change.

5.1.6 The NAF complements the ISO/IEC/IEEE 42010 conceptual model to include enterprises and phases of an enterprise. In this way, architectures can be used to show how they develop and undergo change over time through a process of transformation.

3 Note: Chapter 2 explains analysis of alternatives, trade-off analysis and support for decision making.

6 PURPOSE AND SCOPE OF ARCHITECTURES AND ARCHITECTURE FRAMEWORKS

6.1 Introduction

6.1.1 An architecture may be used to provide a complete expression of any part of the system in an enterprise context. The meta-model defines the essential modelling elements that can be used to describe the system in an enterprise context and its environment. However care must be taken to have a clear purpose in mind for developing any architecture.

6.1.2 Architecture Frameworks may define a common language-independent and tool-independent formalism for architecture representation, and it provides the means to help achieve better communication between architects as well as between architects and stakeholders.

6.1.3 The use of standardized viewpoints serves as a lingua franca as it provides a unified way of describing complex real world objects. It is important both to architects and stakeholders that those involved in an architecture process are aware of this fact and use it to their common interest. This common language will also help to establish a common arena for discussing architectures and consequences across communities of interest in NATO as well as across Nations and organizations.

6.1.4 The NAF supports capturing the vision of the enterprise in all its dimensions and complexity of system-of-interest. The NAF architectures developed will be an important contribution to ensure that the stakeholders of an enterprise are focused on the same goals; development of operational capabilities and the transformational process to reach the objectives of any organization. For illustration, in the defence domain the NATO Federated Mission Networking (FMN) is an example of what NAF architectures will support and in the civil domain an example is the European Air Traffic Management project.

6.1.5 The role of architecture is to provide an abstraction of the real world. By reducing complexity an architecture can be used to support a variety of analyses to address the concerns that the stakeholders have in mind. Many of the required analyses will be performed in specialist tools, informed by the architectures and the analysis results may be used to refine architectures. Some of the key types of analyses that can be supported by an architectural approach include:

Static Analyses – can include capability audit, interoperability analysis or functional analysis.These analyses are often 'paper-based' using simple analysis tools such as database queries and comparisons.

Dynamic Analyses – sometimes referred to as executable models, these analyses typically examine the temporal, spatial, or other performance aspects of a system through dynamic simulations. For example, these analyses might be used to assess the latency of time sensitive targeting systems or conduct traffic analyses on deployed tactical networks under a variety of loading scenarios.

Experimentation – where differing degrees of live versus simulated systems can be deployed during experimentation and there is a high degree of control over the experiment variables. These can be used for a variety of purposes across the acquisition cycle from analysing intervention options to validating new capability prior to its fielding. For example the use of events within NATO such as the Coalition Warrior Interoperability Exercise (CWIX) and experiments held at various battle labs to provide the ability to conduct human-in-the-loop simulations of operational activities can provide venues for experimentation.

Trials – medium to large scale exercises involving fully functional systems and large numbers of personnel, usually conducted in an operational environment as realistic as possible. Such trials are inevitably expensive and are usually only utilized for formal system acceptance or assessment of operational readiness. (Note: Trials can be independently executed or be part of an overall Concept Development & Experimentation (CD&E) process.)

6.2 What is the Value of an Architecture?

6.2.1 Architectures are developed to support strategic planning, transformation, and various types of analyses (i.e., gap, impact, risk) and the decisions made during each of those processes. Additional uses include identifying capability needs, relating needs to systems development and integration, attaining interoperability and supportability, and managing investments. The following describes architecture usage at two different levels[4]:

Enterprise Level – architectures, particularly federated architectures, are used at the enterprise level to make decisions that improve:

- human resource utilization,
- deployment of assets,
- investments,
- identification of the enterprise boundary (external interfaces) and assignment of functional responsibility, and
- structuring the functional activities in terms of projects.

Project Level – architectures are used at the project level to identify capability requirements and operational resource needs that meet business objectives. Project architectures may then be integrated to support decision making at the enterprise level.

6.2.2 Architectures facilitate decision making by conveying the necessary information. Setting architectures within the enterprise context ensures complete, actionable information for more reliable decisions. The following describes architecture data usage for different types of decisions:

Portfolio management – identifies objectives and goals to be satisfied with regards to owned assets (capabilities and systems) and processes to be governed.

Capability and Interoperability Readiness – Assesses capabilities and their implementation (systems, platforms, services and aggregated solutions) against needs and their net-readiness to identify gaps in interoperable features.

Operational Concept Planning – Examines how various mission participants, processes, roles, responsibilities, and information need to work together, to recognize potential problems that may be encountered, and to identify quick fixes that may be available to accomplish a mission.

Acquisition Programme Management and System Development – Expresses the plan and management activities to acquire and develop system concepts, design, and implementation (as they mature over time), which enable and support operational requirements and provide traceability to those requirements. This process must be compliant with the Enterprise objective and operational requirements. It refines operational analysis, performs system analysis, and improves both materiel and non-materiel solution analysis.

Modelling and Simulation – Modelling and simulation techniques can be used in order to assess the business and mission analysis. For example, in the military context through the implementation of mission threads[5] and scenarios[6], thus providing an environment for thorough testing of identified use cases.[7]

4 The NATO EA Policy identifies a third level being the Capability level which is between Enterprise and Project levels.

5 Mission Threads have been described as an operational description of end-to-end activities that accomplish the execution of a mission. No formal definition has been promulgated.

6 A postulated sequence or development of events within a particular setting (Oxford Dictionary).

7 A use case is a term used in systems and software engineering for a list of action or event steps, typically defining the interactions between role (actor) and a system. In systems engineering they are described at a higher level than in software engineering and often represent missions or stakeholder goals.

6.3 Interoperability between Architectures

6.3.1 Architectures must not be produced for the sake of architectures themselves, but as a means to achieve higher level enterprise objectives (i.e. objectives in NATO).

6.3.2 Architecture related processes should be seen as a technique for managing complexity rather than activities to produce models. A common set of architecture processes, such as those specified in NAF, is judged to be the best way of achieving success in the formation of a federation of systems approach.

6.3.3 This concept is not only valid for NATO itself, but also between NATO, Nations and NATO's various partners (Non-NATO Nations, International Organizations (IOs) and Non-Government Organizations (NGOs).

7 NEW FEATURES AND IMPORTANT CHANGES IN NAFv4

7.1 New Features

7.1.1 There are several new features in NAFv4, they include:

- An Architecture Methodology,
- A Grid representation of Viewpoints,
- Adoption of commercial meta-models.

7.2 Architecture Methodology

7.2.1 A new methodology is provided in Chapter 2. This has been developed from accepted best practice to provide:

- Terms and concept for architecting,
- A foundation for architecture activities,
- Architecture principles,
- Architecture activities at enterprise and project levels,
- Architecture repositories and libraries to formalize architecture-based references, allow reuse and improve interoperability between communities.

7.3 Grid Representation

7.3.1 Chapter 3 details the viewpoints that make up NAFv4. These are presented as a grid representation to organize the various subjects of concern (rows) and aspects of concern (columns), logically and consistently to aid architects, as shown below:

Figure 1-1: NAFv4 Viewpoints

					Behaviour					
	Taxonomy	Structure		Connectivity	Processes	States	Sequences	Information	Constraints	Roadmap
Concepts	C1 Capability Taxonomy NAV-2, NCV-2	C2 Enterprise Vision NCV-1		C3 Capability Dependencies NCV-4	C4 Standard Processes NCV-6	C5 Effects NOV-6b		C7 Performance Parameters NCV-1	C8 Planning Assumptions	Cr Capability Roadmap NCV-3
	C1-S1 (NSOV-3)									
Service Specifications	S1 Service Taxonomy NAV-2, NSOV-1			S3 Service Interfaces NSOV-2	S4 Service Functions NSOV-3	S5 Service States NSOV-4b	S6 Service Interactions NSOV-4c	S7 Service I/F Parameters NSOV-2	S8 Service Policy NSOV-4a	Sr Service Roadmap
Logical Specifications	L1 Node Types NAV-2	L2 Logical Scenario NOV-2	L2-L3 (NOV-1)	L3 Node Interactions NOV-2, NOV-3	L4 Logical Activities NOV-5	L5 Logical States NOV-6b	L6 Logical Sequence NOV-6c	L7 Logical Data Model NSV-11a	L8 Logical Constraints NOV-6a	Lr Lines of Development NPV-2
					L4-P4 (NSV-5)					
Physical Resource Specifications	P1 Resource Types NAV-2, NSV-2a,7,9,12	P2 Resource Structure NOV-4,NSV-1		P3 Resource Connectivity NSV-2, NSV-6	P4 Resource Functions NSV-4	P5 Resource States NSV-10b	P6 Resource Sequence NSV-10c	P7 Physical Data Model NSV-11b	P8 Resource Constraints NSV-10a	Pr Configuration Management NSV-8
Architecture Meta-Data	A1 Meta-Data Definitions NAV-3	A2 Architecture Products		A3 Architecture Correspondence ISO42010	A4 Methodology Used NAF Ch2	A5 Architecture Status NAV-1	A6 Architecture Versions NAV-1	A7 Architecture Meta-Data NAV-1/3	A8 Standards NTV-1/2	Ar Architecture Roadmap

7.4 Adoption of Industry Meta-Models

7.4.1 As part of the development of NAFv4 it was agreed that it should make use of commercial architecture meta-models to enable architecting across military and non-military domains. These are described in Chapter 4.

7.5 Architecture Body of Knowledge

7.5.1 NAFv4 will be part of a NATO Architecture Body of Knowledge. The Body of Knowledge will include a number of guides to aid the adoption of NAFv4 such as:

- A complete example of architecture development,
- How to use NAFv4 within NATO to support common architecture tasks such as developing Mission Threads or conducting Capability Planning,
- How to apply the commercial meta-models to develop NAFv4 views,
- Best practice in transitioning from NAFv3 to NAFv4.

Chapter 2 - Methodology

1 FOREWORD

1.1 The NATO Architecture Framework version 4 (NAFv4) is a standard for developing architectures.
1.2 The purpose of this Chapter is to provide a NAFv4 methodology to set up an architecting environment, governing, managing, defining and evaluating architectures.
1.3 The contents of this Chapter should be interpreted as guidance as the level of applicability and tailoring of the NAF methodology will vary according to organization strategy and business/project constraints.

2 SCOPE

2.1 The NAFv4 methodology outlines the approach and the environment in which architecture related activities are performed and architectures are governed, managed, defined and evaluated. This methodology should be tailored by each organization into applicable processes, methods and means relevant to the organization and subject of interest.
2.2 This methodology and the formalism described in Chapters 3 and 4 are to be considered as a constructive generic framework.
2.3 The NAFv4 methodology does not intend to define precisely the terms “Enterprise”, “Organization” and “Project” because the literature provides a lot of definitions for them. However, in this document the meaning is:

Enterprise is where the considered activities take place.

Organization is how the enterprise is organized.

Project is an endeavour to create a system, product or service in accordance with specified resources and requirements.

2.4 Chapter 5 of the NAF includes a glossary that provides specific definitions of terms used in this chapter.
2.5 The methodology addresses the needs of various stakeholders (users, acquirers, providers, builders, etc.) to either develop or use architectures. Three main methodological areas are currently identified:

The architecting at **enterprise level** addresses how a group of people or organizations can work collaboratively on a portfolio of architectures with an enterprise vision. It provides explanation on the architecture landscape with workspace, libraries, and repositories in the enterprise. It also explains how activities can be performed with regards to the enterprise motivation and how activities can be used to govern the enterprise projects.

The architecting at **capability programme/project level** covers libraries, repositories, portfolios and activities used in a capability programme or a project. A project is associated to any architecture within the enterprise.

Foundation for architecting provides prerequisites and value factors to allow the viability of the architectures and their related activities at both the enterprise and the projects.

These are illustrated in Figure 2-1.

Figure 2-1: Three Main Methodological Areas

ENTERPRISE ARCHITECTURE

- Enterprise or strategic scope
- Enterprise motivation data
- Enterprise reference libraries
- Enterprise architecture repositories
- Migration plan for the enterprise transformation
- Portfolios for the enterprise assets
- Enterprise architecture policy
- Enterprise architecting activities

SYSTEM ARCHITECTURE

- Program/project scope)
- Project motivation data
- Project reference libraries
- Project architecture repositories
- Migration plan for the project
- Portfolios for the project assets
- Architecture management plan
- Project Architecting Activities

FOUNDATION FOR ARCHITECTING

- Architecture principles
- Capabilities: means, skills & competencies (tools, disciplines and specialties)
- Patterns for architecture and architecting
- Assets: deliverables and building blocks
- Motivation data for architecting: policies and charters, contracts, gates, readiness and maturity models

Note:
Capabilities Governance with the whole enterprise scope.
Capability management per project.
Artefact description addressed by Enterprise and project

3 WHY DO WE NEED THIS ARCHITECTING METHODOLOGY?

3.1 Based on existing methods and proven experience on architecting through various business domains, this methodology provides a constructive generic framework to ensure efficient architecting. The methods described or referenced in the methodology define the usable and adaptable concepts, means, proceeding and outcomes.

3.2 This methodology provides a foundation to set-up architecting activities within an organization with necessary and justified tailoring to fit with particular architecting context.

3.3 The motivation is to provide a baseline of formalized processes and assets descriptions in order to:

- ease governance and management,
- allow collaborative architecting activities, and
- have unique and homogeneous architecture repository and architecting environment.

4 MAIN CONCEPTS FOR ARCHITECTURE AND ARCHITECTING

4.1 Introduction for Architecting and Architecture

4.1.1 Architecting encompasses the full range of activities of the architect in creating, implementing and managing one or several architectures addressing problems, expectations and/or solutions. The scope related to the architecture generally includes a list of expected capabilities and/or system-of-interest and the enabling systems that sustain the system's viability along its whole life cycle.

4.1.2 The subject of interest may be anything, including a collection of things, analysed with a systemic approach, like an enterprise, a system of systems, a traditional (single) system, a platform, a piece of equipment, a service or a software application.

4.1.3 In many settings, such as product lines, family of systems, programs or enterprises, the architect handles several different architectures at the same time. Architecting aspects include:

- the scope of the architecting effort,
- stakeholder concerns, and
- architecting activities to include producing an architecture description.

4.1.4 In some circumstances, the architect also works on system-agnostic architectures, for example, operational capability definition and mission thread exploration activities. Such architectures are used either to identify systems sustaining the scope of interest or to abstract existing systems in order to explain their provided value.

4.1.5 The architecture of an entity, as defined by ISO/IEC/IEEE 42020[8], is the fundamental concept or properties of an entity in its environment embodied in its elements, relationships, and in the principles of its design and evolution. The architecture expresses:

- the main characteristics of the problem and solution space with possible alternatives. (Note: A complete solution includes the subject of interest and the enabling entities),
- provide orientation data for the processes sustaining the life cycle of the solution related to the architecture,
- the concerns of the Stakeholders for architected entity into formalized views,
- the assumptions made on the environment of each system of the solution to cover the life cycle of the solution (operational processes; natural, human and technical actors interacting with each system; functional and non-functional constraints applied to them: see DLOD[9] PESTEL[10], DOTPMLFI[11] , etc.).

8 ISO/IEC/IEE 42020 Enterprise, Systems and software — Architecture Processes
9 DLOD: United Kingdom Ministry of Defence Lines of Development
10 PESTEL: Political, Economic, Social, Technical, Environmental, Legal (Business Evaluation)
11 DOTMLPFI: Doctrine, Organization, Training, Materiel, Leadership & Education, Personnel, Facilities and Interoperability/Information. See Concept Development and Experimentation Course – Allied Command Transformation 29 Jan – 2 Feb 07, www.dodccrp.org/files/CDE%204-2%20ACT%20CDE%20Process.pdf

5 ARCHITECTING SCOPE

5.1 Introduction

5.1.1 The scope of architecting shall clearly state which part(s) of the lifecycle are being considered out of the entire life cycle of the solution from the earliest concept definition to retirement and possible replacement. This may be by defining specific time periods or phases of the lifecycle.

5.1.2 As long as systems are concerned, discussions of architecting and architectures may occur relative to a subject of interest. Each identified system can also be part of a more extensive system and comprises sub-systems. A notion of a product can also be identified as a system constituent or Architecture Building Block (ABB). Most complex products contain other products (seen within subsystems) capable of independent operation, e.g. a software operating system, with each subsystem having its own architecture.

5.1.3 The scope of architecting encompasses not only technical considerations, but a wide range of developmental, technological, business, operational, organizational, political, economic, legal, regulatory, ecological and social influences, and often aesthetic[12] concerns that influence the solution.

5.2 Stakeholder Concerns, Viewpoints and Perspectives

5.2.1 Stakeholders include customers, designers, users, operators, architects, suppliers, maintainers, accreditors and many actors. Identifying the relevant stakeholders of a subject of interest (e.g. a system, a capability) for each phase of its life-cycle is required to formulate and understand its architecture. A stakeholder may be an individual (e.g. the internal or external identified customer) or a wide-ranging class (e.g. the market demand for this product). Some stakeholders are directly involved in architecting; others can only be concerned or impacted by associated activities or outcomes.

5.2.2 Examples of concerns and impacts are: functionality, feasibility, usage, performance, security, cost, schedule, compliance to regulation. This listing of example concerns gives concrete evidence for the "breadth approach" expressed by Mills Mills, 1985].

5.2.3 An architecture description should be constructed in such a way as to permit separation of concerns through the use of one or more Views constructed in accordance with Viewpoints. An architecture description can be supported by one or several models. Each model may be a part of more than one Architecture View. Models are a way to share information between architecture and views.

5.3 Architecture Dimensions

5.3.1 Several dimensions can be considered for development of architectures. For example:

- architecture life cycle with phases, from creation to closed out. The NAFv4 methodology does not specify the number and names of phases,
- periods of time when architecture applies: from now ("as-is") to a target period ("to-be") and milestones,
- architecture evolution expressed with versions and stages, and
- resource availability including organization and funding.

5.3.2 Architecture viewpoints and perspectives can also be considered as dimensions that transverse the previous ones.

5.4 Kinds of Architectures

5.4.1 The NAF methodology is independent of the various kinds of architectures and architecting styles currently used in industry and governmental organizations.

5.4.2 Nevertheless, different kinds of architectures can be considered according to their purpose, domains of application and roles within entity and architecture life cycles. Architecting may require the use,

12 For example Vitruvius (c. 90-20 B.C.E.) stated that all architectures must satisfy three distinct concerns: firmitas (strength), utilitas (utility) and venustas (beauty).

the development and/or the application of architectures of several kinds. For example, an organization might define kinds of architectures are:

- enterprise-wide architecture descripting the future situation with limited detail. This description normally covers several programs,
- architecture description to be used as reference by a capability/programme or for architecting within a domain, and
- a description limited to the scope of a single project addressing implementation decisions.

Although the term "Baseline Architecture" is often used, this term qualifies an architecture as being a reference for usage rather than being an architecture kind as such. An architecture baseline is an architecture that has been formally agreed and that thereafter serves as the basis for further development. E.g., As-Is (baseline) architecture or baseline technology architecture.

Some other kinds of architectures are also defined in architecture frameworks, like The Open Group Architecture Framework (TOGAF):

Table 2-1 - Kinds of Architecture

TOGAF architectures	Usages
Capability Architecture	An overview of current capabilities, target capabilities, and capability increments to be fulfilled by one or several systems / projects.
Business architecture	A description of the structure and interaction between the business strategy, organization, functions, business processes, and information needs.
Information Systems Architecture	Describing how the enterprise's Information Systems will enable the Business Architecture and the Architecture Vision. Note: An Information system can be seen as a subset of a system.
Technology Architecture	Description of technology assets and standards that are used to implement and realize solutions.

5.5 Architecting Styles

5.5.1 It is widely recognized that the development of an architecting approach is not straightforward and typically the development of an approach is limited by the expertise and experience of an individual architect. This results in varying degrees of success and a continual need to reinvent. To help architects and the problem owners who commission the use, and ultimately control the funding for architecture outputs, a small number of standardized architecting styles have been proposed. These styles help to understand the approach that should be taken; set expectations on what can be achieved; clarify what is involved (e.g. in terms of costs, skills and governance); and, help to understand how value is delivered to the enterprise. The styles are driven by the purpose or reason for the architecture and reflect currently observed best practice.

5.5.2 Four styles of architecting have been identified by architecture practitioners within the United Kingdom (See UK MOD's 'Perfect Storm' – and 'The Need for Architecting Styles', NATO STO-MP-SCI-254 presented at Symposium on Architecture Assessment for NEC, Tallinn, Estonia, 2013 by David Evans and Mike Wilkinson and Niteworks' White Paper, October 2014, Styles of Architecting – A Smarter Approach to Architecting the Defence Enterprise, David Evans). They are as follows:

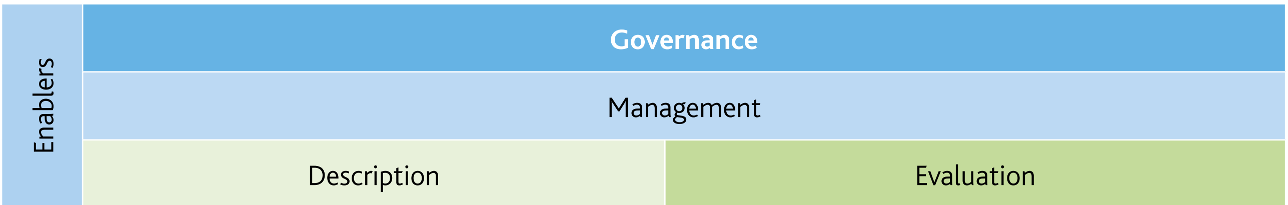
- authoritative,
- directive,
- coordinative, and
- supportive.

5.6 Main Architecture Processes

5.6.1 A first description of process, activities and tasks related to Architecture definition is provided by ISO/IEC/IEEE 15288[13]. A more detailed explanation is given in this section with identification of 5 processes that could be performed by different organizations and projects within an Enterprise.

5.6.2 This description of processes is close to the ISO/IEC/IEEE 42020[14].

Figure 2-2: Architecture Processes



5.6.2.1 Architecture processes can run concurrently, even if the governance and management directions circulate in down-flows and operation reports in up-flows.

5.6.2.2 Architecture description and evaluation are interleaved to regularly state about quality and distance to expectation.

5.6.2.3 The enabling activities are transverse to other architecture processes. They ensure seamless consistency of services and data within the architecting environment.

5.7 Architecture Governance

5.7.1 Governance covers the strategic activities controlling architecture according to enterprise directions and objectives. The main architecture governance activities include:

- establish capability for architecture governance,
- establish strategic desired outcomes for the architecture portfolio,
- evaluate coherency of architecture roadmaps toward desired outcomes,
- provide directions for the architecture portfolio and the related activities,
- monitor the enterprise's portfolio of architectures and the related activities to ensure compliance with the governance directions, and
- decide on necessary corrective actions and iterate.

5.7.2 This process is normally under responsibility of enterprise entities in charge of the consistency of architectures across projects of the enterprise. This consistency concurs to the overall governance of activities and assets of the whole enterprise.

Each activity is governed by principles. The "Design Authority", an external body to the architect team, should be in charge of checking that activities are performed according to these principles.

5.8 Architecture Management

5.8.1 Architecture management is a process to plan, run and monitor architectures along their life cycle. The objective is to have the architectures developed according to enterprise governance direction with regards to stakeholders' expectations.

5.8.2 These activities include:

- establishing capability for management of one or several architectures in the scope of responsibility, and the related activities,
- establishing plans for conducting architecture management activities according to the architecture governance directions,
- providing guidance and direction for architecting activities,

13 ISO/IEC/IEEE 15288 Systems and Software Engineering — System Life Cycle Processes
14 ISO/IEC/IEEE 42020 2016 Enterprise, Systems and Software — Architecture Processes

- monitoring and assess architecture development with management direction, and
- deciding on necessary corrective actions and iterate.

5.8.3 This process is normally lead in different organizations of the enterprise where architecture developments are taking place. It strongly depends on the kinds of architecture being developed.

5.9 Architecture Description

5.9.1 Architecture description process aims to be compliant to ISO/IEC/IEEE 42010[15]. The main activities identified are:

- analyse the problem situation (purpose, scope and objectives),
- identify the stakeholders, their concerns and needs,
- formalize and classify key requirements from collected needs,
- identify the potential solutions,
- identify architecture viewpoints according to stakeholders' concerns,
- develop models and views of candidate architectures from these viewpoints,
- provide the rationale of the potential solutions with regard of requirements and motivation data. In particular, ensure their traceability to motivation data,
- review architecture candidates with stakeholders and get their approval, and
- state relations between candidate architectures and design and other downstream activities.

5.10 Architecture Evaluation

5.10.1 Architecture Evaluation process aims to be compliant to ISO/IEC/IEEE 42020 and ISO/IEC/IEEE 42030[16]. These standards propose architecture evaluation activities including:

- define evaluation purpose, scope and objectives,
- identify the stakeholders of the architecture evaluation, and their concerns or questions,
- determine evaluation criteria (according to stakeholders' concerns/questions) with their relative importance (priorities, weights, etc.),
- determine techniques, methods and tools for performing the evaluation,
- evaluate the architecture,
- collect and understand required information (metrics), and
- formulate the findings and recommendations.

5.11 Architecture Enablers

5.11.1 The purpose of the Architecture Enablement process is to develop, maintain and improve the enabling capabilities, services and resources needed in performing the other architecture processes. This could involve the acquisition or development of these capabilities, services and resources, if needed.

5.11.2 **Enabling capabilities** include, among other things:

- procedures, methods, tools,
- frameworks, architecture viewpoints,
- work product templates,
- decision support systems, storage, and
- configuration management and reference models.

5.11.3 **Enabling services** include, among other things:

- infrastructure, technologies, and
- skilled personnel and automation agents.

5.11.4 **Enabling resources** include, among other things:

- architecture repository, library, registry,
- communication channels and mechanisms,
- human and technical resources, and
- licenses for tools and methods.

---

15 ISO/IEC/IEEE-42010:2011 Systems and software engineering — Architecture description

16 ISO JTC1/SC7/WG2 is working on the project "ISO/IEC 42030 Systems and software engineering – Architecture evaluation" which will provide greater detail on this topic.

5.12 Architecture Life Cycle

5.12.1 An Architecture is a living entity that orientates the life cycle processes (cf. ISO/IEC/IEEE 15288 and 12207) of the architected entity. An architecture has its own life cycle (a beginning and an end when this architecture is no longer applicable or suitable) which orients the life cycle of the architected entity.

5.12.2 Processes (or activities) sustain the subject of interest along its life cycle; i.e. any activity necessary to make this subject viable along its life cycle. When directly associated to a system, the architecture life cycle maps the whole system life cycle from its conception to its disposal. Architecture provides a technical contract to system owners and builders, through an architecture plan, by framing candidate systems and subsystems of interest and associated enabling systems. This includes the critical path from the earliest baseline to its numerous increments, which are handled by appropriate versions of the system engineering management plans.

5.12.3 Sometimes an architecture can express various expectations not directly linked with a single system, or their life-cycle. For example:

- **Architecture issued prior to identification of system(s)** describes the problem space, to allow solving the problem according to stakeholders' concerns. In this case, only business/operational views and capability views are elaborated. They are used to update the doctrine, operational processes, or to acquire and govern systems or services. The architecture life cycle starts when problem analysis starts, and finishes when both the problem and solution spaces are no longer concerned,
- **Architecture issued to cover several projects worked concurrently** along a period of time: it may be called overarching architecture and the set of projects are considered as a programme. The architecture life cycle starts with the beginning of the programme and ends with the last project,
- **Architecture issued to cover several systems/products worked concurrently** along a period of time: Product lines, families of systems and systems of systems are belonging to this case. The architecture provides an overall definition which is normally refined by individual system/product architectures,
- **Architecture issued to cover several projects worked in sequence** – when possible – along a period of time: In this case, the architecture provides the transformation roadmap, including systems/products evolution and/or replacement, to fulfil architecture objectives at the considered period of time.

5.12.4 These examples highlight the need to customize architectural environments, activities and outcomes in order to be fit for purpose. Customization will also depend on the enterprise organization and the complexity of both problem and solution, which can call for different plans and activities on the architected entity.

5.13 Architectures and Architecting Activities in the Enterprise

5.13.1 Considering an enterprise as a group of people or a group of organizations, most of the time, the enterprise business is divided into units, domains and projects involving all the necessary disciplines and expertize.

5.13.2 An enterprise can consist of enterprises within it. In that case the inner enterprises are acting within their own business processes and within the overall enterprise business according to several possible models being federated, cooperative and collaborative.

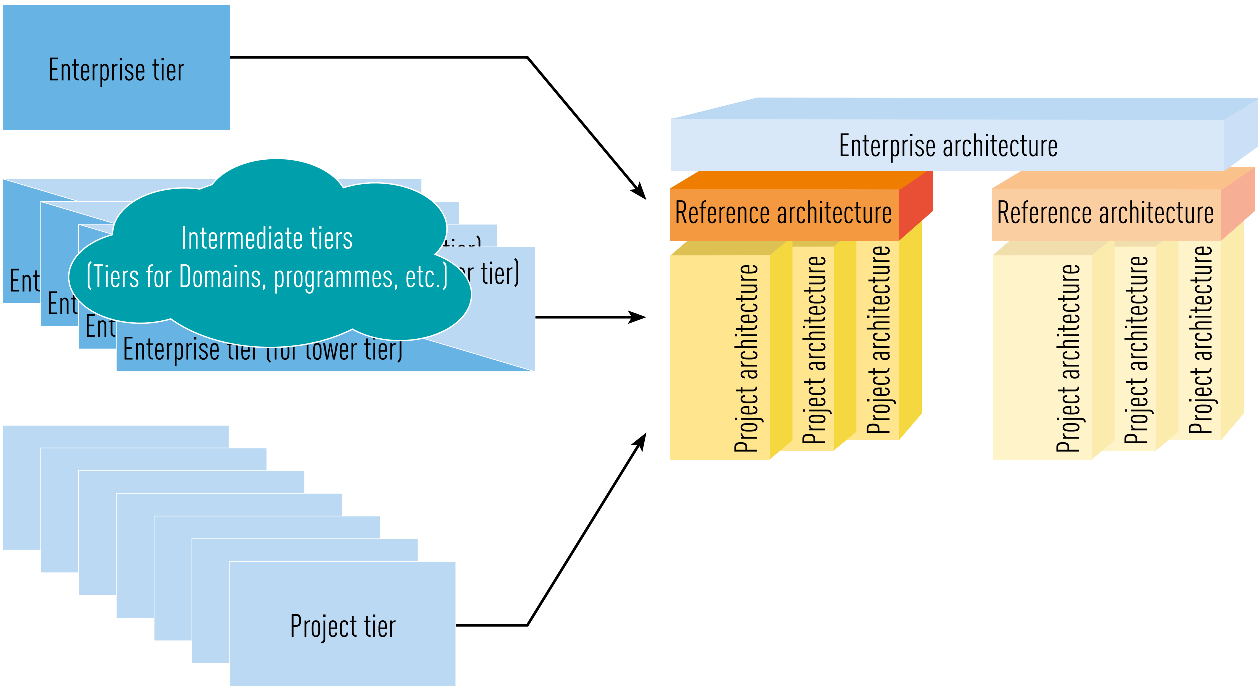
5.13.3 Architecture activities have to be considered at any enterprise level and architecture entity since each is expected to work with a systemic approach, i.e. each enterprise entity acquires and/or develop systems and/or products to cover its own usage and for its deliveries.

5.13.4 Within these enterprise entities, each work unit can be considered as a project. This project can be performed either entirely in a relevant enterprise entity, with other enterprise entities, or with third-parties. The architecture and related activities can be seen as being at a project level when the project is performed by a single entity or when there is no interest by stakeholders to know how the project is completed from a given analysis point of view. Architecture and related activities for the enterprise scope can be performed by several enterprise entities according to several organizations: collaborative architecture activities, multi-tier (or multi-layer) sequential activities, multi-tier concurrent activities, etc.

5.13.5 For multi-tier architecting activities in an enterprise, the middle-tiers act as Project for the upper tier and as Enterprise for the lower tier. This means that an architect or a team can work within a double architecture environment. However, the two roles and environments have to be clearly distinguished in order to achieve clear outcomes and interaction between the levels.

5.13.6 The following figure provides an example about how to map the multi-tier architecture activities with the examples of different kinds of architecture.

Figure 2-3: Example of Multi-Tier Architecture Activities



- Enterprise Architectures are developed by the enterprise tier activities,
- Reference architectures are developed by domain and programme tier activities,
- And system architectures are developed at project levels.

5.14 Architecture Framework

Architecture Framework TOGAF v9.1, page 45]: "is a foundational structure, or set of structures, which can be used for developing a broad range of different architectures. It should describe a method for designing a target state of the enterprise in terms of a set of building blocks, and for showing how the building blocks fit together. It should contain a set of tools and provide a common vocabulary. It should also include a list of recommended standards and compliant products that can be used to implement the building blocks."

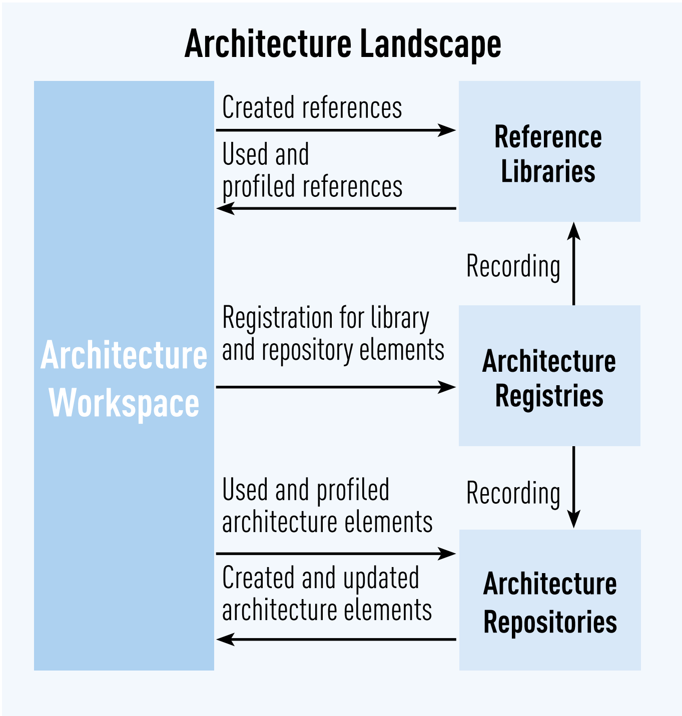
No architecture framework is currently fully compliant with the above definition. Some frameworks focus on architecture description, while others are more oriented to process description. Very few include tools and/or standards.

Part of an architecture framework is related to architecture domain with reference standards and products. This part is to be defined and adjusted in line with the enterprise organization and policies.

5.14.1 **Architecture Framework as Working Environment**,

- An architecture framework should be used as a working environment. This environment is called an 'architecture landscape'.

Figure 2-4: Architecture Landscape



- The architecture landscape is structured in 4 main areas:
  - the architecture workspace where architectures are developed,
  - the reference libraries containing any information useful for the architects to either do their job or to get architecture related information,
  - the architecture repositories where architectures and architecture building blocks are made available:
    - to be used as references for implementation.
    - to provide principles and guidelines for development of other architectures and elements, and
  - the architecture registries record the usage of elements in reference libraries and architecture repositories in order to allow their management and governance,
- Architecture landscapes can be considered at any tier of the Enterprise performing architecting activities or accessing architecting outcomes: whole enterprise, domains, programmes and projects.

Right-to-know and relevance of information will be considered for each architecture landscape.

5.14.2 Enterprise Architecture Landscape

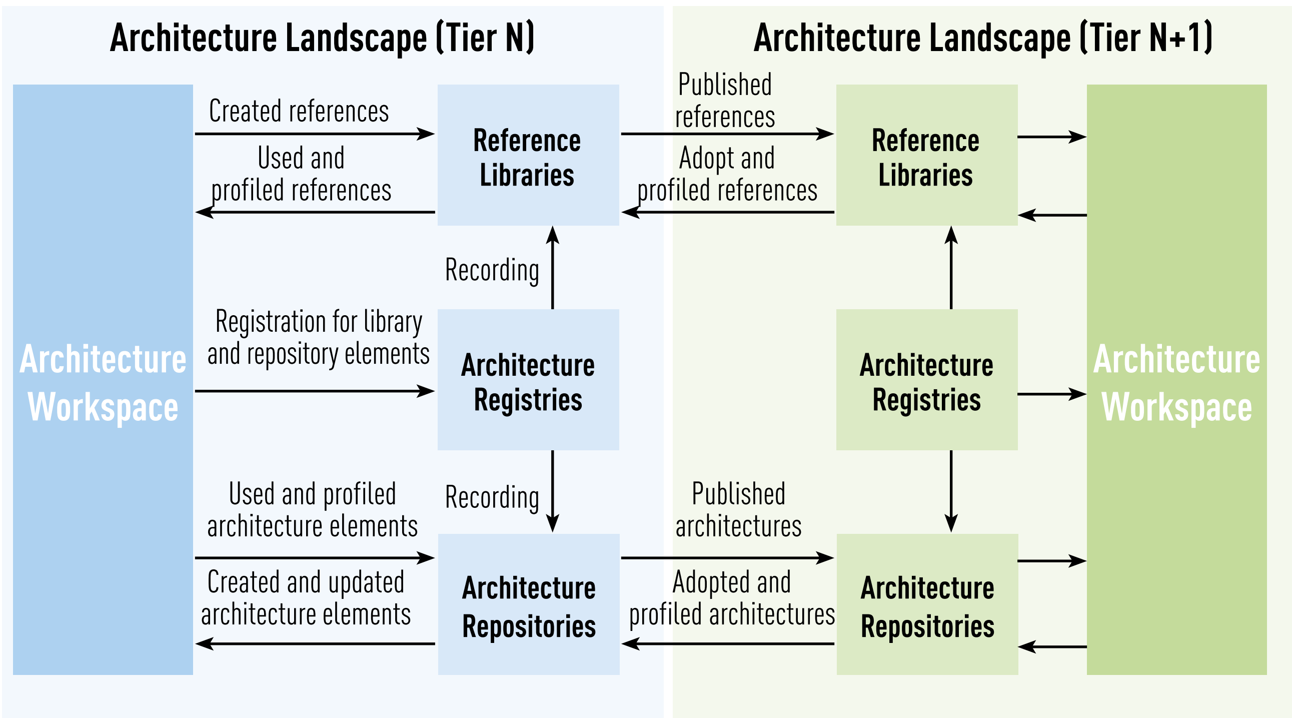
- It allows enterprise architecting activities in the enterprise to cover multi-programme, multi-project and enterprise-width business,
- Enterprise reference libraries and Enterprise Architecture repository host data being available for the other stakeholders of the enterprise. In these shared spaces, data elements are stored within baselines, i.e. the data elements are recorded according to their temporal and structural dependability. A baseline is characterized by a given time and a data configuration,
- Enterprise reference libraries host the baselines of assets reusable by any architect of the enterprise,
- Enterprise Architecture repositories host the baselines of the architectures and architecture elements produced or updated by any architects of the enterprise, and approved by the board of architects,
- The Enterprise Architecture workspace is the environment where the architects act at the enterprise level. This area contains work-products and data developed by architects prior to their publication as a new or updated reference, architecture element and architecture,
- Enterprise Architecture registries record the usage of elements of reference libraries and of architecture repositories in the Enterprise Architecture landscape.

5.14.3 Project Architecture Landscape

- This landscape has exactly the same structure as an Enterprise Architecture landscape:
    - project reference libraries host the baselines of assets reusable by the architects in a project,
    - project architecture repositories host the baselines of architectures and architecture elements produced or updated by architects of the project,
    - project architecture workspace is the environment where the architects work for the project. This area contains any work-product and data developed by architects prior to their publication as new or updated references, architectures and architecture elements, and
    - project architecture registries record the usage of elements in reference libraries and architecture repositories in the project's architecture landscape.

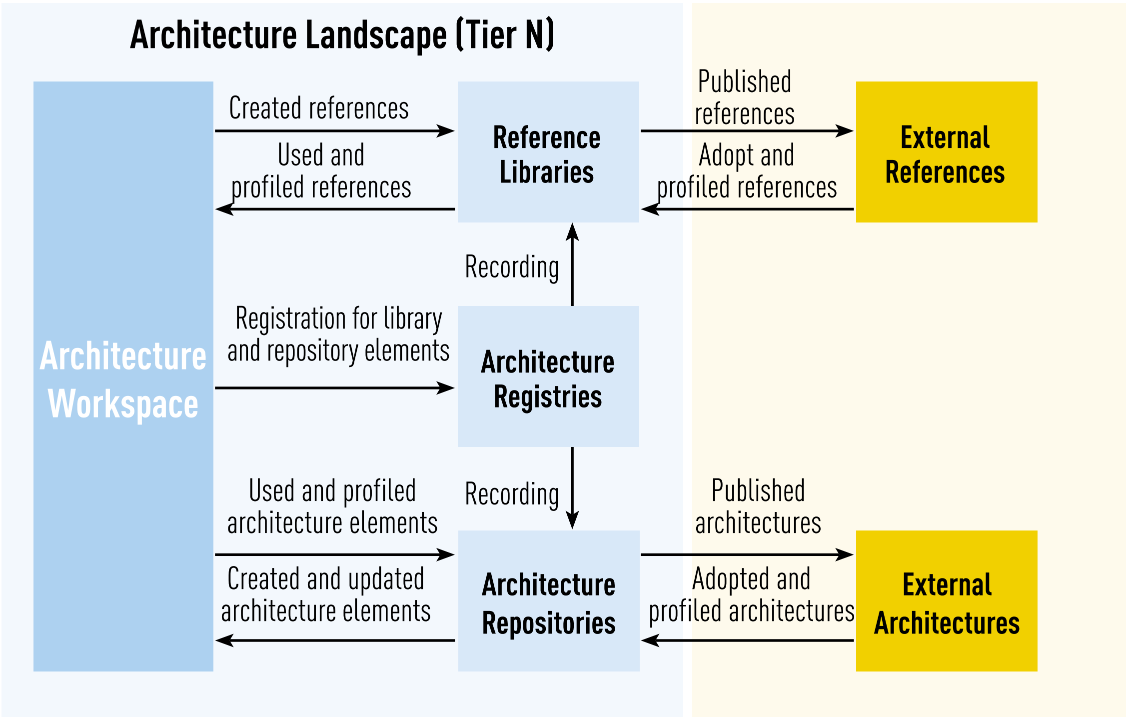
5.14.4 Architecture Landscape Interactions

Figure 2-5: Architecture Landscape Interactions (View from Tier N)



- Interactions occur between architecture landscapes when multi-tier architecting activities are in place in an enterprise (See Figure 2-3 with the example of enterprise, domains, programmes and projects tiers). Architecture landscapes are complementary structures. Considering the interaction from one tier point of view:
  - the architecture landscape exposes usable or mandatory data (references and architectures) for the other tiers, and
  - the architecture landscape uses and profiles data elaborated by the other tiers.

Figure 2-6: Architecture Landscape External Interactions



- Architecture landscapes also interact with the enterprise environment to:
  - collect external data elements enriching the enterprise' assets with references, architectures and architectures elements (with respect to the copyrights and licenses), and
  - publish enterprise assets (with respect to the right-to-know).

5.14.5 Reference Libraries

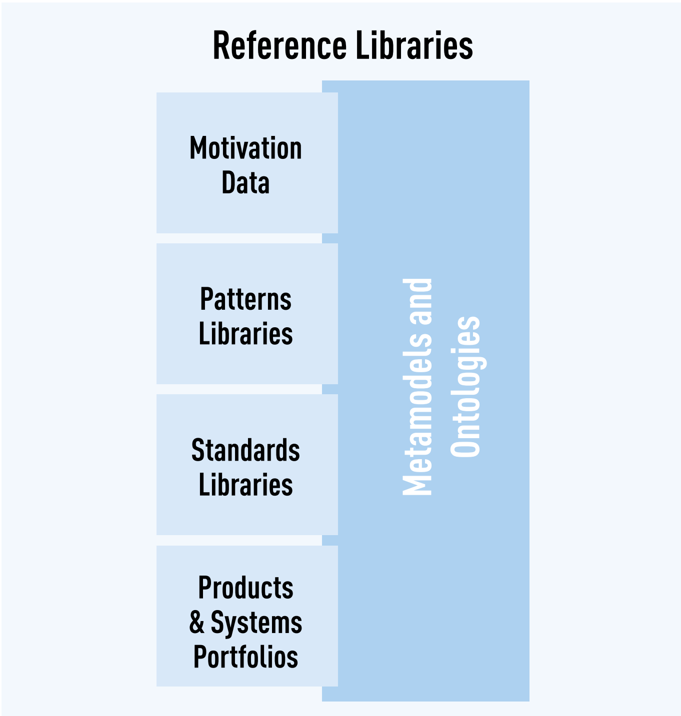
- Reference libraries host the baselines of assets reusable by architects in their activities per architecting organization. This information can:
  - either come from the lower architecting tiers in the enterprise organization, in which case the consistency and the relevance for the current tier is checked, or
  - be created and/or collected for lower tiers through architecting activities.

The reference libraries may include:

- meta-models and ontologies providing the terms and concepts used in the reference system. This information provides the enterprise the foundations to build the vocabulary of the projects. They can be updated and augmented by projects-specific terms and concepts,
- customizable architecture motivation data. Architecture motivation data could cover the concepts defined in The Open Management Group Business Motivation Model (see Figure 2-9) with:
  - information directing or defining the business aspirations: business vision, goals and objectives,
  - the means to realize the business aspiration: missions and course of action,

- the stakeholders' value system and associated assessment elements: key requirements, risks, opportunities, cost and value per viewpoint, and assessment criteria and key questions.
- business directions and guidance for activities.

Figure 2-7: Reference Libraries



A more detailed description of architecture motivation data is given in "Architecture Repositories", the next section.

- patterns providing canonical templates, constructs and activities,
- standards, de facto (standards issued from best practices or enterprise policies) and de jure (standards issued from professional, governmental or international regulatory bodies) references, and
- portfolios of products (including services) and systems, or more generically building block (e.g., locations, organizations, process, information products that are recommended for usage in the architecture activities.

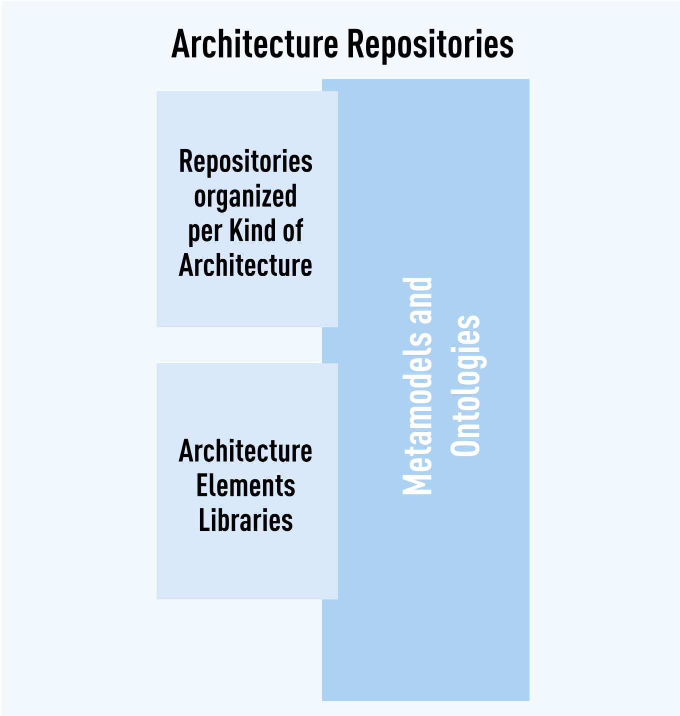
5.15 Architecture Repositories

5.15.1 Architecture repositories host the baselines of architecture elements produced or updated by architects per architecting organization.

5.15.2 Architecture repositories include:

- the different kinds of architectures,
- the architecture elements: architecture patterns and architecture building blocks as borrowed from reference libraries, or created for the purpose of the architecture to be developed, and
- meta-models and ontologies formalizing the terms and concepts used in the architecture repositories.

Figure 2-8: Architecture Repositories



5.16 Architecture Motivation Data

5.16.1 Architecture motivation data gathers information and references relevance for initialization of architecture, orientation of architecting activities and analysis of findings.

5.16.2 Motivation data includes the problem vision, goals and objectives to be met by the architecture. From these aspirations, the organization identifies the main concerns subject to questions along architecting activities. Statements of missions communicate the direction of the organization intending to pursue the vision. A strategy (i.e. long term plan) defines how to achieve corresponding goals.

5.16.3 Architecting activities are oriented by external and internal drivers and rely on well-defined criteria to assess the findings. Drivers may impact the use of reference processes and may call for architecture method tailoring.

5.16.4 For instance, when interoperability drives architecting, the method recommends to tackle business and/operational concerns prior to any migration activity. According to architecting policies, architects will plan the evaluation of alternatives to actual architecture operational products to meet objectives.

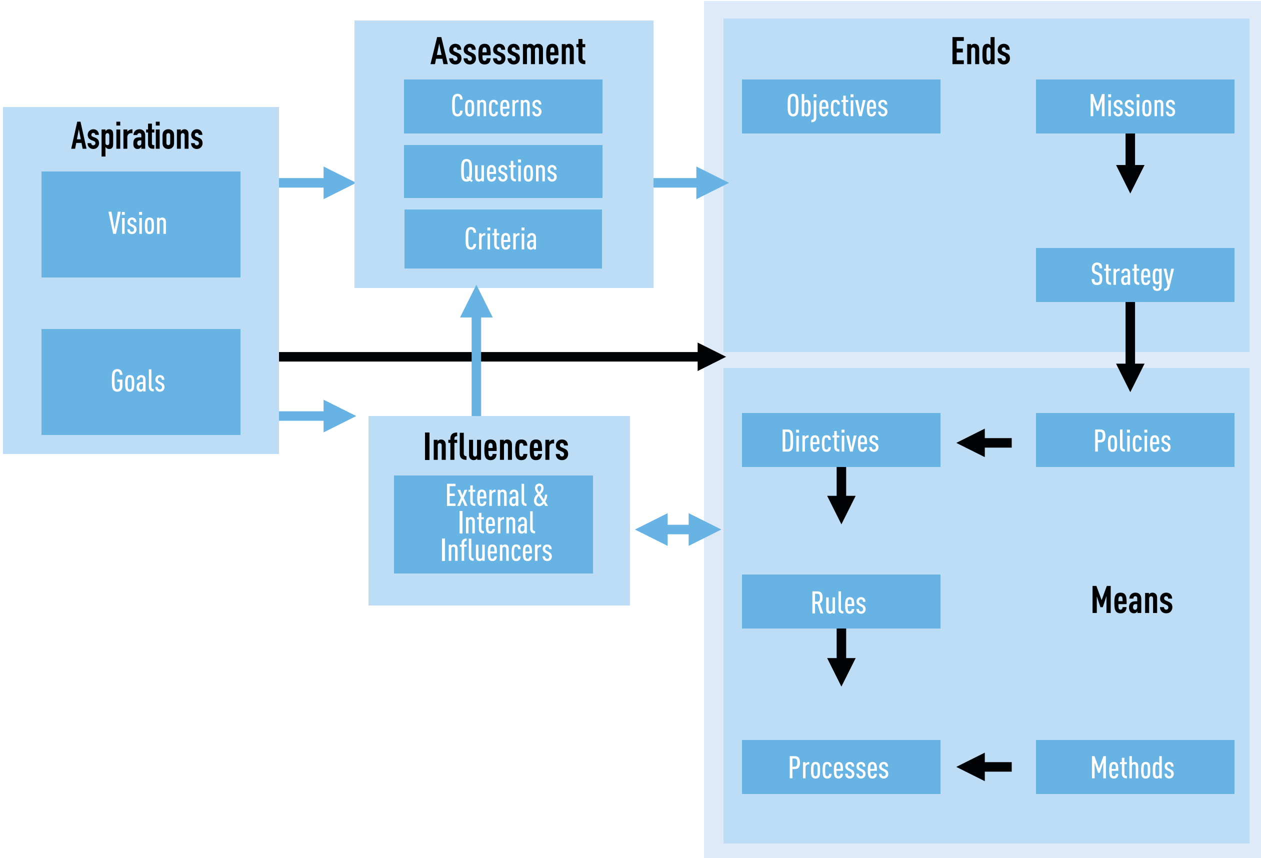
5.16.5 Policies and rules set the context of process adaptation to major architecture drivers such as interoperability.

5.16.6 The main interfaces to engineering processes (reference documents, engineering change requests, checks) are specified in architecture policies, including guidance rules to align with enterprise and projects policies.

5.16.7 The Business Motivation Model Version 1[17] defines the relationships between various motivation elements. These relationships are shown at Figure 2-9.

17 OMG Document: formal/2008-08-02

Figure 2-9: Motivation Data



5.17 Manage Architecture Motivation Data

5.17.1 Architecture motivation data is a living entity initialized by an architecture change request and fed by the architecture landscape that led to change approval. It includes different types of data:

- **Contextual Data** – business elements (business model, directives, eco-system analyses, product portfolios, project portfolios, architecture principles, assumptions for architecture governance and management, norms and standards, including export control and regulations),
- **Justification Data** – architecture change justification and impact analyses,
- **Orientation Data** – architecture policy, approved architecture vision that specifies business goals, expected timeline and the right capabilities to meet the goals at the right time,
- **Planning Data** – architecture statement of work and plans (governance, management, configuration management, resources). The architecture plans will follow one of the architecture driver set (e.g. DLOD, PESTEL and DOTPMLFI) as agreed by stakeholders).

5.17.2 Architecture workflows are conceived to revisit motivation data according to the findings of previous stages in terms of:

- Eevolution of context and/or need,
- new scenarios, same or new missions, for the same or different context, requiring the same or different quality of service,
- to deliver in the same or different timeline,
- evolution of norms/standards/regulations: update or obsolescence of (domain, technology, business, political, societal) norms,
- concept change: doctrine, business domain and technology,
- enterprise strategy change (product-line, roadmap, partnership, acquisition policies). DLOD: Defence Lines of Development, and
- markets, stakeholders, organization, enablers, products, roadmaps, compliance to customer requirements or product line approach, etc.

5.17.3 The most important principle for architecture change decision is to achieve stakeholder agreement on priority, over expected capabilities from business, on capability and technical standpoints. The second principle that architects will observe is checking consistency of capability dependency models with capability phasing views to highlight capability critical dependencies, taking into account:

- agreement on priority of expected capabilities from evolution timeline and related metric evolution viewpoint,
- stakeholder's agreement on weight of each criterion used to assess and compare alternatives of architecture,
- revisiting (baseline of) stakeholders' requirements according to priority and weights of criteria, and
- revisiting motivation data according to outputs of the last iteration of the vision stage.

5.18 Architecture Policy

5.18.1 An architecture policy is a set of principles guiding architecture decisions and achieving rationale outcomes. It has a title, is owned by an authority acting to govern the architecture activities, and includes the architecture glossary.

5.18.2 Architecture policies are adopted by the board of architects and implemented in procedures and/or protocols to be applied by architects when performing their activities.

5.18.3 An architecture policy will assist architects in defining the scope and boundaries of architecture products, setting interfaces to architecture resource and facilities, and to subsequent engineering processes and activities.

5.18.4 In order to plan consistent and affordable roadmaps of architecture activities and work products, the architecture policy includes the principles to interact with:

- Building Block Owners,
- Support Entities,
- Experts and Specialists,
- Strategists, and
- Decision-Makers.

5.19 Architecture Management Plan

5.19.1 This plan provides the overall framework for architecture development. The goal is to deliver the appropriate guidance to support acceptance, while ensuring that architecture models are exploited to reuse assets and support efficiently test cases. It describes:

- The **architecting strategy** according to enterprise policies: architecting activities to run, expected product's focus to reach architecture goals as stated in the corresponding state of work.

- **Tailoring of architecting** iterations and architecture products to reach architecture goals. It includes a stop criteria for each planned activity.

- **Architecture landscapes**, within and outside the enterprise, as described in sections 5.14.2 to 5.14.4:
    - reference libraries hosting reusable assets, including reference skills, methods, and tools to achieve activities,
    - repositories hosting baselined architecture products,
    - workspaces hosting architecture development data and work products,
    - interaction between landscapes along architecture life cycle, and
    - interaction between architecture activities and other activities (planning, engineering, operations and maintenance).

- The **planning of activities** and control of architecture requirements and products.

- The **governance and management** processes of architecting activities.

5.19.2 The architecture management plan is a living document. It is updated as much as necessary to reflect changes, especially, changes of goals, landscapes and their interactions.

5.20 Migration Plan

5.20.1 Migration to an agreed future architecture is planned and described taking into account the scenarios allowing handling critical dependencies to other projects, if any. The plan recalls the context and scope of migration to the baseline and describes:

- the main goals from stakeholder perspectives,
- reference policies and rules for migration including conflict resolution principles and configuration management rules,
- if necessary, the migration strategy and criteria,
- roles and responsibilities to manage the migration process in alignment with reference policies,
- migration timeline and decision making policy, and
- migration means: motivation data, library, repository and dashboard.

5.21 Evaluation Report

5.21.1 Identified alternatives of architecture are evaluated according a selected set of criteria, reflecting the main concerns of and agreed with stakeholders. The evaluation report describes the following points:

- scope of evaluation,
- description of evaluated alternatives,
- evaluation objectives and criteria,
- evaluation method and rationale,
- evaluation results, and
- interpretation of results and recommendations.

5.21.2 Recommendations are provided to support decision making; decisions concern the approval of alternatives and of proposed trade-offs, where necessary. Trade-offs will usually concern the negotiated non-functional properties to keep architecture in line with budgets and timeline, though evaluated timeline and/or value-to-cost may suggest transitioning via more affordable solutions to target.

5.22 Main Architecture Document

5.22.1 The main architecture document provides the overall landscape is initialized from the current landscape. It recalls architecture context, goals and objectives and synthesizes the findings of architecting activities.

5.22.2 It defines the architecting method and associated principles, and provides a rationale for customization based on agreed drivers, internal and external. The rationale includes an explanation of concerns and criteria selected to meet architecture objectives.

5.22.3 Principles usually include the expected number of alternatives and the criteria to distinguish clearly between each alternative (a property, a capability level). Properties include architecture availability, characteristics and cost (development migration, application costs).

5.22.4 The body of the main architecture document describes retained architecture alternatives from stakeholder's viewpoints, and for each candidate, the set of assumptions and results interpreted to support decision-making.

5.22.5 The executive summary of the main architecture document provides a synthesis of:

- stakes, constraints and assets enabling to approach the vision,
- principles and criteria to shortlist alternative of architectures, and
- criteria to find the best candidate or to propose a trade-off from shortlisted candidates.

5.23 Architecture Dashboard

5.23.1 Architecture dashboard synthesizes data needed to monitor architecting activities until architecture goals are considered as achieved or, until a decision to suspend part or whole of monitored activities is taken by the architecture board.

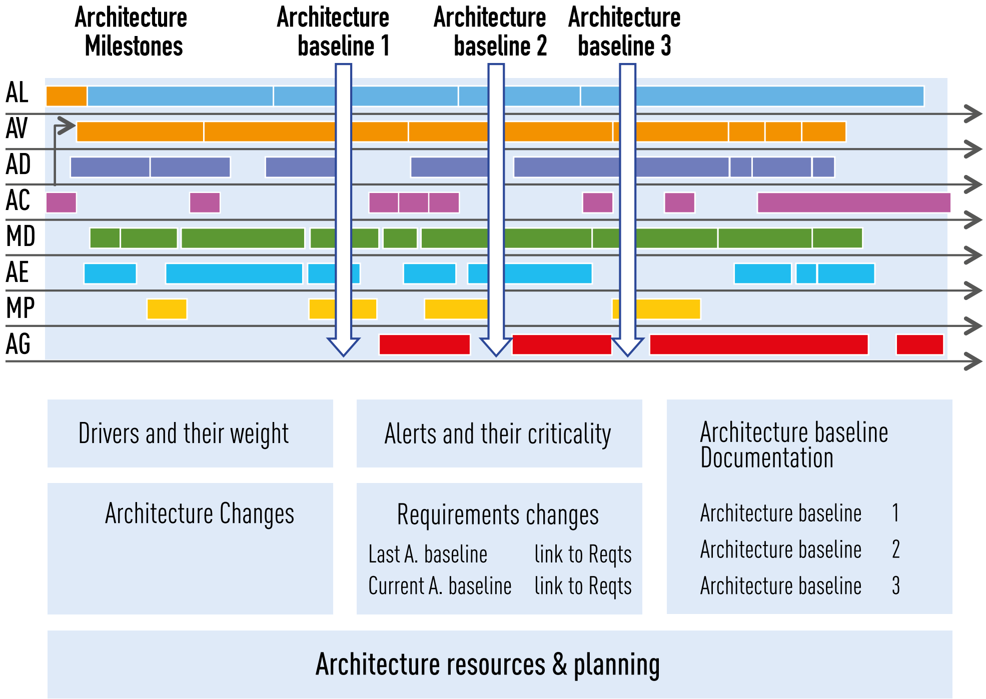
5.23.2 Architecture has its own life cycle. The dashboard highlights architecture key milestones as they are agreed at initialization/update of architecture vision, in consistency with enterprise directives and policies.

5.23.3 Architecture life cycle is different from projects milestones. However, projects plans include synchronization points to align with architecture evolution.

5.23.4 Two kinds of milestones can be distinguished in a dashboard:

- milestones for architecture products to be developed and evaluated by architects: we call them hereafter Architecting Milestones, and
- milestones for architecture to be developed and implemented by projects: We call them hereafter Architecture Milestones.

Figure 2-10: Dashboard Depicting Interleaving Activities along an Architecture Life



5.23.5 Architecture Milestones correspond to capability configurations of the selected architecture trade-off solution to fit customer and user expectations:

- capability levels: operational relevance, deployment readiness, integration with legacy are examples of architecture milestones from a customer perspective,
- technical feasibility, with respect to standards, norms and laws (international and or local) can lead to different configuration milestones from the designer perspective,
- roadmaps of building blocks of interest induce milestones from development perspectives, and
- technology readiness roadmaps dictate milestones from technology readiness perspective.

5.23.6 Architecting milestones correspond to the phases and timelines to deliver architecture products and propose trade-offs. They must conform to the architecture management plan (enterprise/ project).

5.23.7 Therefore, a dashboard may be parameterized to monitor activities run along architecting phases of an architecture project and the evolution of architecture baselines as managed within an enterprise portfolio.

5.23.8 Each goal might be refined along architecting phases into sub-goals and associated intermediate milestones. Each of them allows running analyses while composing logically and/or physically (when concept experiment is part of the evaluation process), selected building blocks and sub-systems of the architecture libraries with remaining part of the solution. Analyses consider architecture qualities, performances, human factors and any property aiming to satisfy operational needs.

5.23.9 Architecture goals, together with the Landscape and Architecture Milestones form the core of the architecture motivation data and shall be consistent with the architecture management plan.

6 ARCHITECTING ACTIVITY

6.1 Architecting Stages

6.1.1 Figure 2-11 describes architecting activities in an architecting organization. They are organized in 8 stages, as follows:

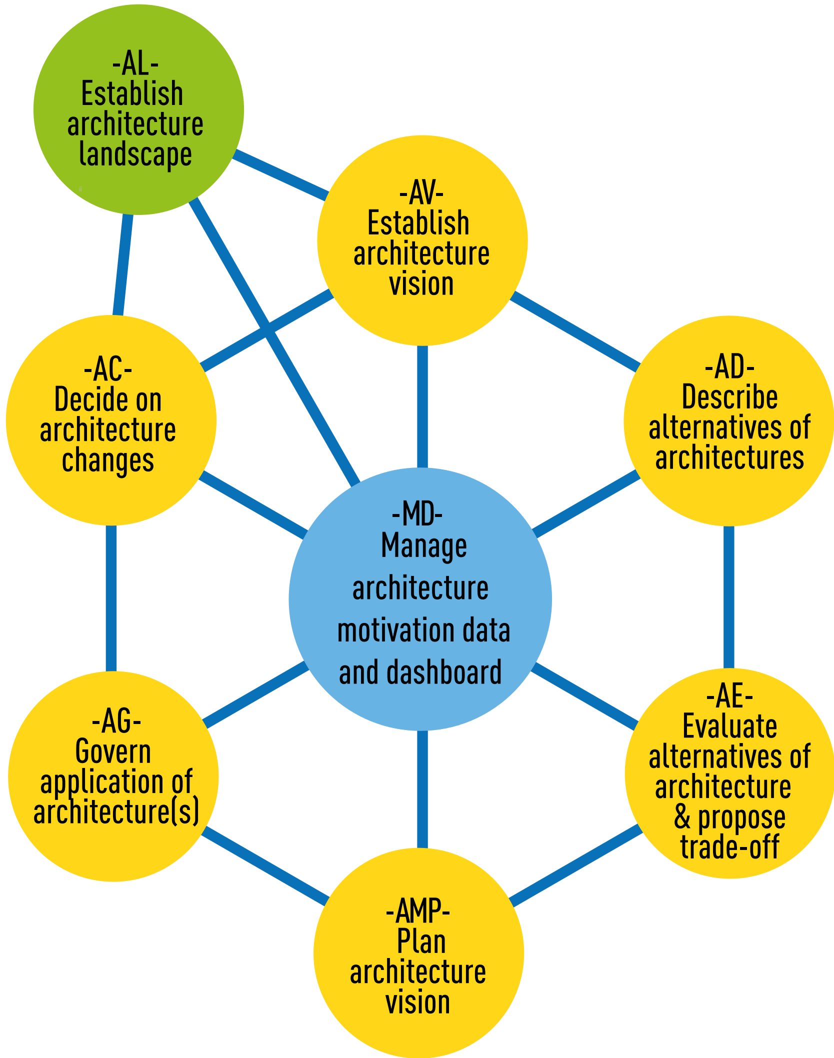
Table 2-2 - Architecting Stages

	Stages	Description
1	Architecture Landscape (AL)	Describes the overall context and defines the capabilities and means to develop an architecture.
2	Architecture Vision (AV)	Defines the architecture vision taking into account the landscape, stakes and time to market (or time to Customer).
3	Architecture Description (AD)	Describes architecture from stakeholders' viewpoints according to landscape, and identify a set of alternatives of architectures for evaluation.
4	Architecture Evaluation (AE)	Updates architecture evaluation criteria set in motivation data to evaluate each alternative, identify the best ones, and elaborate change requests allowing to build the best trade-off from approved best alternatives.
5	Plan Migration (PM)	Updates architecture migration plan and provides rationale for application.
6	Architecture Governance (AG)	Checks the application the best architecture trade-off according to the migration plan and provide guidance to resolve dependency conflicts.
7	Architecture Changes (AC)	Elaborate and get approval on requests for architecture change.
8	Motivation & Dashboard (MD)	Manages architecture context, constraints and drivers and provide views on architecture progress status and dependencies to other architectures and building blocks, through a dashboard aligning products with landscape (reference libraries and repositories).

6.1.2 The method is inspired by the architecture description method of The Open Group Architecture Framework / Architecture Development Method (TOGAF/ADM), however it is different, in order to:

- comply with evolving architecture standards (ISO/IEC/IEEE 42010, draft of ISO/IEC 42020 and 42030),
- ease its deployment within various contexts, not only information technology, and
- allow flexibility in the navigation through architecting stages.

Figure 2-11: Architecting Stages



6.1.3 **The Method**:

- allows the use of any number of Viewpoint(s) and Views per architecting stage,
- aims to capture and manage architecture motivation data, i.e. any element that will steer architecting activities from architecture vision to architecture baseline. This will extend the traditional requirement baseline with goals, expectations, constraints, drivers, risks, costs, value and opportunities. Therefore, while requirements are at the core of the TOGAF/ADM, the NAFv4 method extends the TOGAF/ADM requirement management stage and includes traceability of architecture products. This is used for defining and maintaining an architecture dashboard,
- allows more emphasis on the decision to change architecture and re-orientate the architecture due to a major evolution of motivation data, and
- provides guidance on architecture assessment and trade-offs analyses using motivation data (stakes, objectives, constraints) which can lead to different criteria and techniques for identification and comparison of alternatives.

6.1.4 Each alternative of architecture is described by artefacts (architecture products) of benefit to the stakeholders, which are aligned to architecture requirements. This includes functional and non-functional requirements and an architecture roadmap aligning with capability increments.

6.1.5 Evaluation of architecture alternatives is performed against criteria such as cost, operational effectiveness, system performances, system qualities and time to capability milestones. These criteria are usually expressed by customers or deduced from market analysis.

6.1.6 The DoDAF[18] architecture process, described in Figure 2-12, can be mapped to following stages of the NAFv4 methodology:

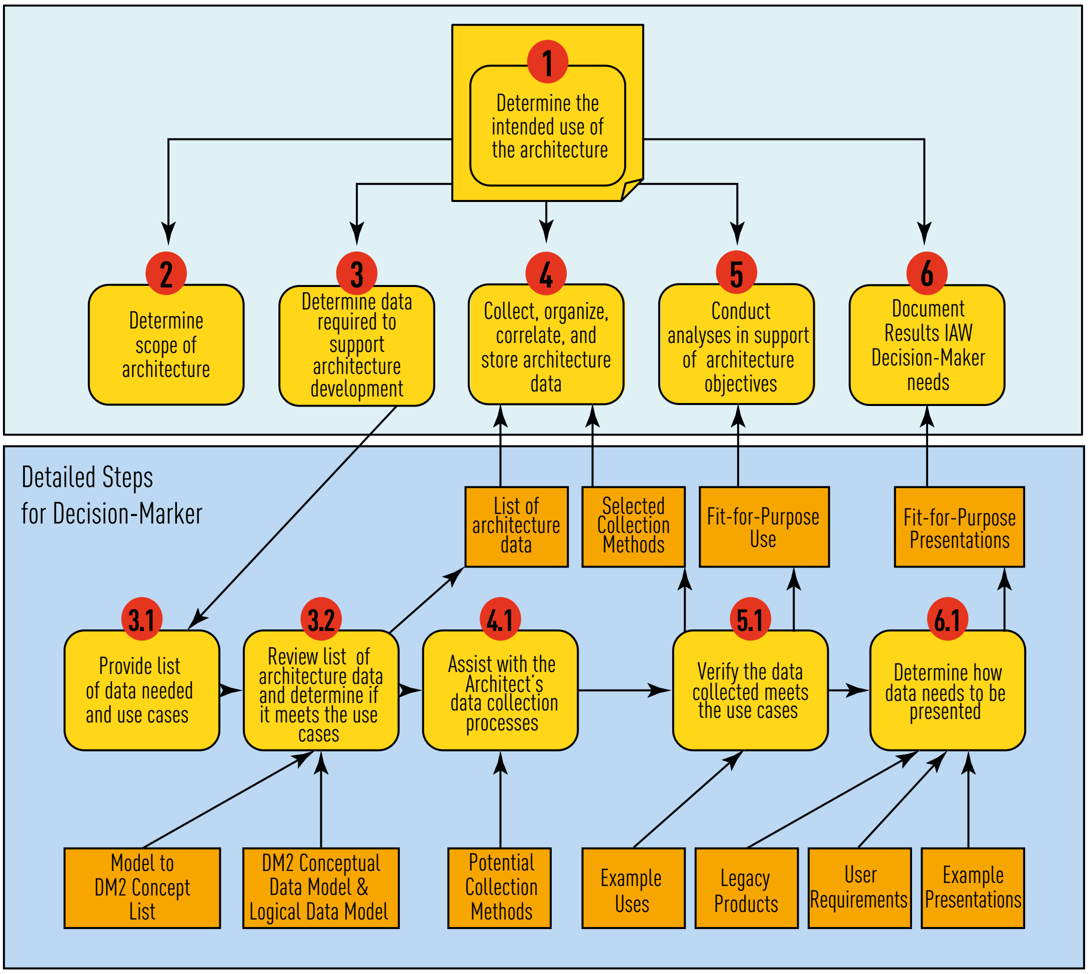
- Establish project architecture landscape,
- Manage architecture motivation data (scope, objectives, policies, requirements, etc.),
- Establish architecture vision,
- Describe alternatives of architecture,
- Evaluate alternatives of architecture.

6.1.7 The NAFv4 methodology defines eight stages see Figure 2-11, visited iteratively to support architecture decision making to deliver an architecture baseline. Each stage has objectives. It refines architecture and creates artefacts based on artefacts created from previous iterations, and from any source of problem and solution contexts. A prerequisite to any iteration of the NATO Architecture Methodology for architecting will be agreement on:

- Scope and level of abstraction,
- Timeline, milestones (progress, validation),
- Stop criteria,
- Acceptance criteria.

6.1.8 The method is compliant with the Six-step process for architecting introduced by DoDAF (See Figure 2-12). It extends this process to establish migration plans towards new architecture reference and candidate target architectures, and govern implementation projects in consistency with enterprise portfolios (e.g. product portfolios and libraries of standards).

Figure 2-12: Six-Steps Architecture Process DoDAF v2.0]



6.2 Architecting dynamics

6.2.1 Along architecture life cycle, architecting activities are grouped in consistent stages that can be orchestrated in different schemes; some activities can be repeated and several iterations involving specific stages may be necessary to reach architecture goals.

6.2.2 Objectives and plan of each phase are key inputs to the dashboard. Architects plan stages and define success criteria collected in the architecture motivation data. The architecture management plan captures justified cycles, iterations and synchronizations with other tiers architectures.

6.2.3 Additional information if any (criticality, priority, weighting) on success criteria are usually submitted for approval of the governance board along trade-offs activities.

6.2.4 Figure 2-13 provides some examples of architecting iterations:

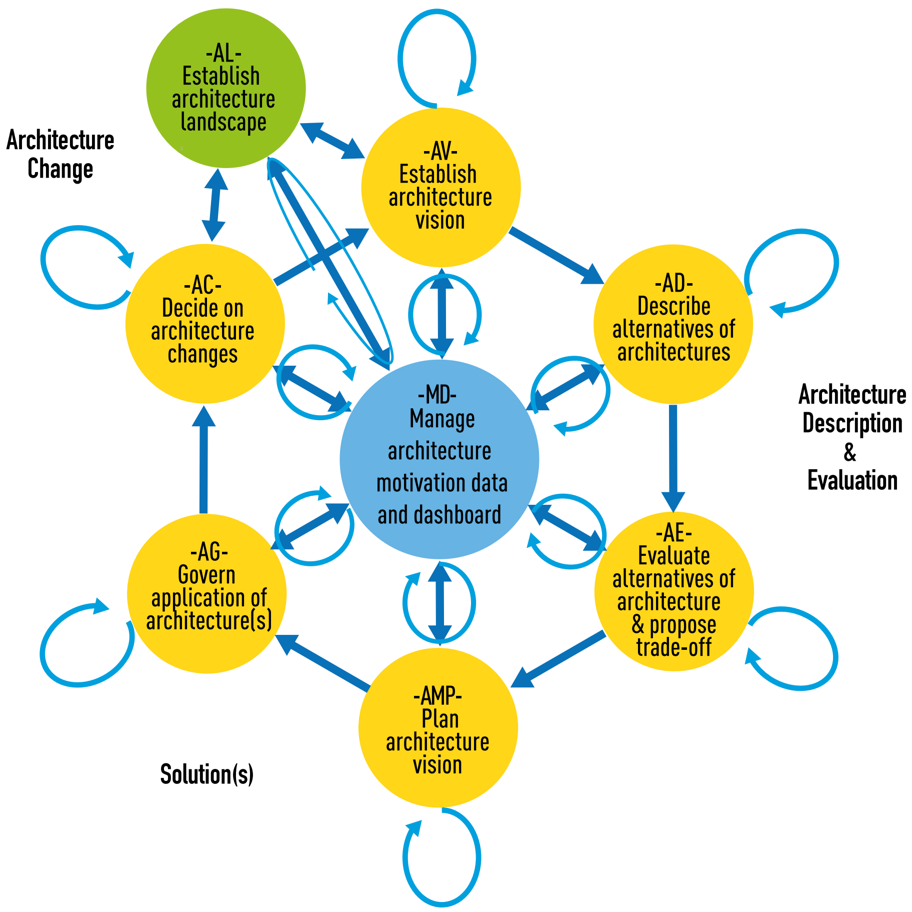
- **Iteration around stages**: The completion of a whole cycle of architecture work may be necessary to set rapidly a broad scene of architecture changes and impacts, to refine through further iterations,
- **Iterating between stages**: The neighbours of a given stage may be revisited to refine the findings of preceding stages as depicted in Figure 2-11 e.g. returning to 'Description of Architecture' on completion of 'Evaluation of Enterprise Architecture' to describe a trade-off between the most promising alternatives). Two other kinds of iterations may be noted:
  - Between 'Migration planning' and 'Governance of application of architecture',
  - Between 'Architecture change' and 'Architecture vision'.
- **Iteration around a single phase**: Stage description supports repeated execution of the activities within a single stage, e.g. a number of iterations of architecture description of architecture to establish consistent architecture products from multiple viewpoints.

6.2.5 At each stage, activities can use and update motivation data (see iteration around motivation data). Approved updates are used to update the dashboard, where necessary.

6.2.6 There are many drivers for tailoring the architecture dynamics: maturity, policies and complexity:

- the vision can be agreed by stakeholders at first iteration when business is not new for them. Otherwise, more iteration may be necessary to reconcile stakeholders' expectations in the vision,
- the level of maturity of product/technical architecture can call for enforcement or lightening of activities at architecture description stage,
- enterprise principles such as product-line policies may shorten the space of possible alternatives to reach business goals,
- the status (evolution, diversity, lack) of standards and norms may lead to more or less alternatives, whether to sustain architecture with regards to standards forecast or to reduce the space of alternatives for non-compliance of the product line to the target business, and
- complexity of organization as established at landscape (interleaving projects, architecture critical dependencies) can call for more or less complex principles to maintain a coherent architecture dashboard.

Figure 2-13: Architecting Cycles & Iterations



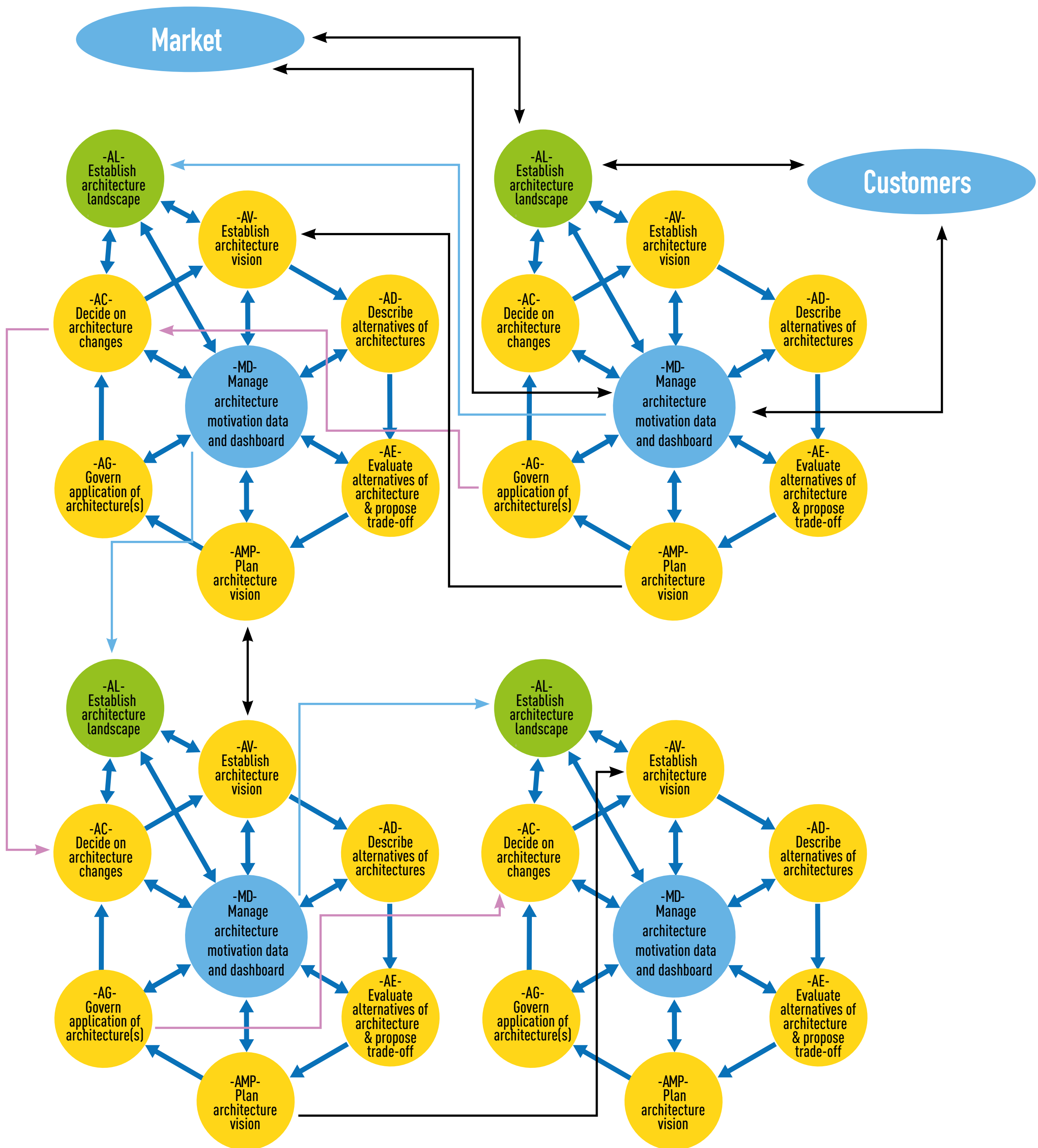
6.3 Multi-tier Architecting

6.3.1 Architecture activities can be run by different tiers: the enterprise, domains within the enterprise, and programmes in enterprise domains, projects, belonging to or shared by programmes or portfolios.

6.3.2 The architecture environment has to consider therefore target markets, customers and shareholders policies, as depicted in Figure 2-14. Architecture changes driven by markets and or customers trigger vision updates at enterprise tier, whilst transformation will be managed and checked at different domains, starting from updates to their vision. Programme and project visions are impacted accordingly.

6.3.3 Landscapes are updated from enterprise down to projects, and from projects up to enterprise, to enable overall governance of enterprise transformation.

Figure 2-14: Architecting Environment



6.3.4 In Figure 2-14:

- blue lines impact/enrich/adapt motivations at domain level inspired by markets/customers analyses/RFI,
- black lines indicate impact of migration planning on current local Vision, and
- purple lines highlight dependency between governing implementations and "decide architecture change" at domain, and/or programme and /or project level.

7 ARCHITECTING FOR THE ENTERPRISE SCOPE

7.1 Introduction

7.1.1 Architecture elaborated to master the overall enterprise business are typically:

7.1.2 Architecture of the enterprise itself. The enterprise is therefore analysed with a systemic approach from the enterprise internal and external stakeholder's viewpoints. This allows formalizing the enterprise processes, roles, information system(s), assets, etc.

7.1.3 Architectures used by the programmes and the projects of the enterprise in order to deliver the enterprise systems/products required by internal and external contracts.

7.1.4 In both cases these architectures provide directions and guidance for the enterprise programmes and projects in charge of developing and maintaining either the enterprise itself or the enterprise systems/ products.

7.1.5 These architectures are considered an input for enterprise governance.

7.2 Overview of the Enterprise Architecting Stages

Table 2-3 – Overview of the Enterprise Architecting Stages

	Stages	Description
1	**Enterprise**: Architecture Landscape (AL)	Put in place the Enterprise Architecture context with identification of the stakeholders, and definition the organizational context, architecture principles, capabilities, processes, outcomes, roles and responsibilities.
2	**Enterprise**: Architecture Vision (AV)	Get an updated Enterprise Architecture vision with related stakeholders, key-requirements and constraints, architecture management plan, relevant activities and outcomes.
3	**Enterprise**: Architecture Description (AD)	Define the Enterprise Architecture viewpoints according to the concerns of the stakeholders and provide an approved set of alternatives of Enterprise Architectures.
4	**Enterprise**: Architecture Evaluation (AE)	Define the evaluation criteria according to the concerns of the stakeholders, evaluate each alternative of Enterprise Architectures, and get an approved selection among the alternatives of Enterprise Architectures for application and possibly request for evolution.
5	**Enterprise**: Plan Migration (PM)	Get an updated transformation roadmap for application of the Enterprise Architecture with a rationale and a governance model.
6	**Enterprise**: Architecture Governance (AG)	Check for the application of the Enterprise Architecture according to the migration plan and provide recommendation.
7	**Enterprise**: Architecture Changes (AC)	Decide on the requests for change, evaluate the level of applicability of the Enterprise Architectures and decide if iterations are needed to update the Enterprise Architectures.
8	**Enterprise**: Motivation & Dashboard (MD)	Put in place a selection of data and build a dashboard reflecting the motivation of the stakeholders. Maintain the reference libraries and architecture repositories to be in line.

For governance activities, it is highly recommended to consider COBIT[19] and ISO 38500[20] in addition to NAF Chapter 2.

19 COBIT (Control Objectives for Information and related Technology): COBIT 5 is a framework for IT governance provided by the Information Systems Audit and Control Association (ISACA).

20 ISO/IEC 38500 Information technology -- Governance of IT for the organization.

7.3 Enterprise Architecting Activities

Table 2-4 – Enterprise: Architecture Landscape (AL)	
Objectives	**Task**
• To formalize the organizational context where the Enterprise Architecture activities take place. • To identify the stakeholders of the Enterprise Architectures and their related activities, with their expectations. • To define the constraining Enterprise Architecture principles. • To define the Enterprise Architecture process with roles, responsibilities, work-products and workflow. • To define the capabilities for Enterprise Architecture work. • To get a commitment on the Enterprise Architecture process and usage of its outcomes.	• Identify the sponsors and the stakeholders for the enterprise Architectures. • Formalize the architecture principles and process consistently with the enterprise directives and the other enterprise processes. • Establish the Enterprise Architecture landscape. • Define the technical and human capabilities for architecture work: methodologies, tools, skills and competencies, etc. • Establish Enterprise Architecture team and organization. • Validate architecture principles, process and capabilities with the Stakeholders.
Inputs	**Outputs**
• Enterprise strategy, policies, direction and guidance. • Enterprise motivation model: business principles, business goals, and business, driver, etc. • Agreement on NAF usage, with possibly some other working references.	• Enterprise Architecture processes and the associated organizational model for Enterprise Architecture activities, with definition of workflows and roles. • Tailored NATO Architecture Framework, including Enterprise Architecture principles. • Usable Enterprise Architecture landscape breakdown structure, including libraries and repositories. • Rationale for compliance to enterprise motivation data (business principles, business goals, business drivers, etc.). • Assumption for architecture governance and management.
Recommended Views	**Stakeholders**
A1 to A7.	• The expectations related to the Enterprise Architecture activities are provided by the stakeholders, i.e. any people having concerns about the Enterprise Architecture related activities. • The Enterprise Architecture landscape is proposed by the Board of Enterprise Architects. • Outputs are agreed by the Enterprise Architecture Governance Board.

Table 2-5 – Enterprise: Architecture Vision (AV)	
Objectives	**Task**
For a particular cycle of architecture activities: • To review the list of the stakeholders for the architected entity, • To formalize and update the key-requirements and constraints from the architecture stakeholders, • To get the updated architecture vision, • To plan the architecture activities to be performed for the architecting cycle, • To check the coherency by other Enterprise Architecture activities on other Enterprise Architectures and other possible parallel architecture cycles, • To get approval to the architecture management plans and outcomes	• Identify the stakeholders for this cycle, with their concerns, and key-requirements. • Confirm or update the Enterprise Architecture principles, • Check and update the enterprise business motivation data against these key-requirements, • Develop and update the architecture vision (key-views) per main stakeholder viewpoints, • Estimate the impact on the enterprise transformation plan: risks, cost, value and opportunities, • Develop Enterprise Architecture management plans and statement of architecture work, • Review the architecture vision and plans with the stakeholders.
Inputs	**Outputs**
• Request for the Enterprise Architecture evolution, • Enterprise motivation data, • Organizational model for Enterprise Architecture, • Pre-existing Enterprise Architecture vision, • Enterprise Architecture landscape.	• Updated approved architecture vision. • Approved plans and statements of work. • Updated architecture principles. • Updated enterprise motivation data.
Recommended Views	**Stakeholders**
• A3, Ar, • C5, • Cr, Sr, Lr, Pr, • C1, S1, L1, P1, A1, • A2, L2-L3 (Architecture Context Diagram (ACD)), L2, C2.	• The expectations regarding the Enterprise Architectures are provided by the stakeholders, i.e. any people having concerns about the Enterprise Architectures, • The Enterprise Architecture vision is proposed by the Board of Enterprise Architects, • Outputs are agreed by the Enterprise Architecture Governance Board, • Executive Management, • Board of Directors.

Table 2-6 – Enterprise: Architecture Description (AD)	
Objectives	**Task**
• To validate the viewpoints with respect to their concerns of the stakeholders, • To provide one or several alternatives of description for an Enterprise Architecture through these viewpoints, • To get an agreement of the alternatives of Enterprise Architectures.	• Analyse the description objectives from the Enterprise Architecture vision, • Refine the list of stakeholders and their concerns with regards to the enterprise motivation data, • Provide rationale for each choice of alternatives, • Refine the architecture viewpoints from the architecture vision for the alternatives, • Perform gap analysis between the Enterprise Architecture vision and the Enterprise Architecture description, • Check the Enterprise Architecture landscape for the architecture description, • Select, describe or update the relevant architecture views according to the viewpoint and concerns, • Trace the architecture views against the enterprise motivation data elements, • Finalize and review the Enterprise Architectures with the stakeholders, • Create architecture definition document for this iteration.
Inputs	**Outputs**
• Request for architecture work with a statement of work, • Enterprise Architecture vision (list of stakeholders, concerns, viewpoints, Architecture overview), • Enterprise motivation data, • Architecture principles, • Pre-existing Enterprise Architecture description in the Enterprise Architecture repositories, • Enterprise Architecture landscape.	• Reviewed described alternatives for the Enterprise Architectures, • Traceability between the Enterprise Architecture views and enterprise motivation data elements, • Architecture definition document, • Gaps with regards to Enterprise Architecture vision (and proposed evolutions).
Recommended Views	**Stakeholders**
• C1 to 8, Cr, S1 to 8, Sr, L1 to 8,Lr, P1 to 8, Pr, • A1, A2, L2-L3 (ACD), A8.	• The concerns related to subjects covered by Enterprise Architectures are provided by the stakeholders, i.e. any people having concerns about the targets and impacts of Enterprise Architectures, • The Enterprise Architecture description is proposed by the Board of Enterprise Architects, • Outputs are agreed by the Enterprise Architecture Governance Board.

Table 2-7 – Enterprise: Architecture Evaluation (AE)	
Objectives	**Task**
• To formalize the evaluation criteria according to the concerns of the stakeholders, • To evaluate each candidate Enterprise Architecture, • To evaluate the risk, cost, value and opportunities for each Enterprise Architecture, • To select the Enterprise Architectures for application.	• Define the evaluation objectives from the Enterprise Architecture vision, • Refine the list of stakeholders, their concerns and questions with regards to the enterprise motivation data, • Define the evaluation criteria from the concerns of the stakeholders, with their relative importance (priorities, weights, etc.), • Determine techniques, methods and tools for performing the evaluation, • Evaluate each architecture alternative with collection and understanding of required information (metrics), • Formulate the findings per architecture alternative, • Perform trade-off analysis with estimate of risk, cost, value and opportunities, • Choose the best alternatives of Enterprise Architectures with rationale against the enterprise motivation data, • Perform gap analysis between the evaluation objectives and the achieved architecture evaluation, • Finalize and review the Enterprise Architecture evaluation results. • Request for change of the alternatives of architectures as necessary, • Create architecture evaluation document for this iteration.
Inputs	**Outputs**
• Request for architecture work with a statement of work, • Enterprise Architecture vision (list of stakeholders, concerns and questions), • Enterprise motivation data, • Architecture principles, • Pre-existing Enterprise Architecture, evaluation elements in the Enterprise Architecture repositories, • Enterprise Architecture landscape, • Enterprise Architectures descriptions.	• Reviewed selection of Enterprise Architectures with assessment of risk, cost, value and opportunities, • Architecture evaluation document including objectives, criteria, evaluation results and selection, • Gaps with regards to Enterprise Architecture vision (gaps with the evaluation objectives), • Requests for changes of the alternatives of architectures.
Recommended Views	**Stakeholders**
• C1 to C8, Cr, S1 to 8, Sr, L1 to 8,Lr, P1 to 8, Pr, • A1, A2, L2-L3 (ACD), A8.	• The evaluation criteria related to subjects covered by Enterprise Architecture are provided by the Stakeholders, i.e. any people having concerns about the targets and impacts of Enterprise Architectures, • The evaluation report is proposed by the Board of Enterprise Architects, • Outputs are agreed by the Enterprise Architecture Governance Board.

Table 2-8 – Enterprise: Plan Migration (PM)

Objectives	Task
• To get updated a roadmap for enterprise projects which progressively apply the architectures, • To demonstrate that enterprise transformation satisfies the enterprise motivation data, • To provide a governance model for application of the Enterprise Architectures.	• Analyse the transformation objectives from the Enterprise Architecture vision, • Identify individual projects, with work-products, timing, effort and resources. • Prioritize the migration projects through the conduct of the enterprise business model validation, • Build an enterprise transformation roadmap showing how projects implement Enterprise Architecture through phases and increments, • Assess the roadmap with cost, benefits, risks and opportunities, • Create the enterprise transformation plan and review it with the stakeholders, • State on the evolution of the Enterprise Architectures.
Inputs	**Outputs**
• Request for architecture work with a statement of work, • Enterprise Architecture vision (list of stakeholders and concerns, transformation outline), • Enterprise motivation data (including policies and rules for transformation), • Architecture principles, • Pre-existing entreprise transformation actions, • Enterprise Architecture landscape.	• Enterprise roadmap, • Enterprise transformation plan, • Portfolio of enterprise projects, • Architecture contract per project or programme, • Change requests for Enterprise Architectures.
Recommended Views	**Stakeholders**
• Cr, Sr, Lr, Pr, Ar, • C8, S8, L8, P8, A8, • C3, • Mapping of Lr over Cr.	• The Enterprise transformation plan is proposed by the Board of Enterprise Architects, • Outputs are agreed by the Enterprise Architecture Governance Board.

Table 2-9 – Enterprise: Architecture Governance (AG)	
Objectives	**Task**
• To ensure correct application of the Enterprise Architectures in the enterprise transformation, • To provide recommendation towards the governance authority of the enterprise transformation.	• Establish directives and guidance for governance of the application of the Enterprise Architectures, • Monitor the application of enterprise transformation through reviews of the enterprise projects, organized the governance authority of the enterprise transformation, • Evaluate the gaps of application with regards to the enterprise transformation plan, • Direct the application by corrective recommendation given to the governance authority of the enterprise transformation, • State on the evolution of the Enterprise Architectures.
Inputs	**Outputs**
• Request for architecture work with a statement of work, • Enterprise Architecture vision (governance outline), • Enterprise motivation data (including policies and rules for transformation), • Enterprise transformation plan, • Portfolio of enterprise projects. • Architecture contract per project or programme.	• Governance model (directive and guidance) for application of the Enterprise Architectures, • Corrective recommendation for applications of the Enterprise Architectures, • Change requests for Enterprise Architectures.
Recommended Views	**Stakeholders**
• A1 to A8, Ar, • C1, C2, S1, S2, L1, L2, L2-L3, P1, P2.	• The governance model is proposed by the Board of Enterprise Architects, • Outputs are agreed by the Enterprise Architecture Governance Board.

Table 2-10 – Enterprise: Architecture Changes (AC)	
Objectives	**Task**
• To transform the requests for changes into decisions for changes in the Enterprise Architecture landscape, Enterprise Architectures, architecture principles and enterprise motivation data, • To decide on the level of applicability of the Enterprise Architectures, • To decide on the need to iterate for one or several Enterprise Architectures (stop criteria).	• Analyse the requests for changes with regards to the current Enterprise Architecture vision and enterprise motivation data, • Perform impact analysis of the Enterprise Architecture landscape, Enterprise Architectures, architecture principles and enterprise motivation data, • Define needs for update architecture principles and the enterprise motivation data, • Define needs for evolution of Enterprise Architecture landscape, • Define needs for a new iteration for evolution of one or several Enterprise Architectures.
Inputs	**Outputs**
• Change requests for Enterprise Architectures, • Enterprise motivation data, • Organization model for Enterprise Architecture, • Enterprise Architecture vision, • Enterprise Architecture landscape.	• Needs for evolution of Enterprise Architecture landscape, • Needs for evolution of one or several Enterprise Architectures, • Needs for updated architecture principles and change request for the enterprise motivation data evolution.
Recommended Views	**Stakeholders**
• A5, A6, A7.	• The needs for evolution are proposed by the Board of Enterprise Architects, • Outputs are agreed by the Enterprise Architecture Governance Board.

Table 2-11 – Enterprise: Motivation & Dashboard (MD)	
Objectives	**Task**
• To manage a consistent access to the enterprise motivation data. • To provide consistent architecture dashboard related to activities, Enterprise Architecture landscape (including Enterprise Architectures in repositories) and enterprise resources.	• Manage updates of the enterprise motivation data asked by the Enterprise Architecture Governance Board and those coming from the Enterprise Architecture stages, • Analyse enterprise external and internal architectures and architecture elements able to enrich the enterprise Architecture repositories. Update the repositories, as necessary, • Analyse enterprise external and internal references able to enrich enterprise reference libraries. Update the libraries, as necessary, • Monitor the performance of architecture related activities with regards to inputs and output dependencies, work requests, usage of human and technical resources and Enterprise Architecture landscape, • Manage a consistent access to enterprise motivation data, • Report to the Enterprise Architecture Governance Board.
Inputs	**Outputs**
• Enterprise request for update of the enterprise motivation data, • Enterprise external and internal architectures and architecture elements, • Enterprise external and internal references, • Organizational model for Enterprise Architecture, • Enterprise Architecture landscape.	• Request for update of Enterprise Architecture landscape, • Updated enterprise motivation data, • Report to the Enterprise Architecture Governance Board.
Recommended Views	**Stakeholders**
• A1, A7, A5.	• The enterprise motivation data are proposed by the Board of Enterprise Architects, • Outputs are agreed by the Enterprise Architecture Governance Board.

8 ARCHITECTING IN A PROJECT

8.1 Overview of Project Architecting Activities

8.1.1 Project architecture defines the rationale for architecture moving from the "As-is" to a "To-be" architecture. Starting from the overall context, and applying enterprise directives and policies, the project vision is set according to the concerns of stakeholders and associated priorities. The latter are used to initialize key architecture requirements as part of the motivation data. During architecting activities, the motivation data is enriched consistently with the rationale associated to identified architecture alternatives, evaluation criteria and trade-offs when necessary.

8.1.2 Evaluation criteria are initialized from vision elements, namely architecture objectives.

8.1.3 The description stage identifies and describes alternatives of architectures which satisfy key architecture requirements and known constraints.

8.1.4 The evaluation stage provides support to decision-making, using criteria agreed by stakeholders.

8.2 Project Architecting Activities

<table>
<tr><th colspan="3">Table 2-12 – Project: Architecture Landscape (AL)</th></tr>
<tr><th>Objectives</th><th colspan="2">Task</th></tr>
<tr><td>• Establish the architecting capability according to expectations and context, scope and target,
• Tailor and get stakeholder’s agreement on the data that will guide architecture activities:
- Enterprise directives on architecture.
- Enterprise principles applicable to architecting.
- Infrastructure, methods, tools and principles enabling activities from architecture vision to architecture definition.
- Enterprise principles monitoring progress of architecture.</td><td colspan="2">• Confirm enterprise expectations, map to project motivation data and set corresponding indicators in the dashboard,
• Define architecture team members, their personal and collective roles objectives to fulfil the architecture capability,
• Per identified role: collect, analyse and mark architecture inputs to feed the motivation data and dashboard,
• Define architecture workflows: link main roles, outcomes and communication policies. In particular, specify roles and workflows defining the interface to Enterprise Architecture level (Architecture governance board),
• Set infrastructure, select and adapt tools and method supporting architecture capability,
• Initialize architecture repositories to manage architecture,
• Set architecture principles to apply by architecture project actors,
• Define architecture dashboard to monitor architecture progress,
• Check consistency of project rules with enterprise principles.</td></tr>
<tr><th>Inputs</th><th colspan="2">Outputs</th></tr>
<tr><td>• Context, drivers and constraints calling for architecture capability:
- Business strategy, product-line strategy, portfolios, partnerships and contract agreements,
• Architecture scope and expectations, in terms of business objectives and timeframes,
• Resources plan to sustain architecture capability along the agreed architecting timeframe (i.e., from vision to new baseline),
• Principles and constraints from enterprise business motivation data,
• Architecture state of work,
• Architecture documents of legacy systems: interfaces, life cycles, known constraints,
• Architecture management plan outline.</td><td colspan="2">• Organization of architecture team: architecture OBS and agreed workflow from vision to architecture baseline,
• Tailoring of the architecting process to enable the workflow,
• Definition of resources, skills and roles according to the tailored architecting process,
• Definition of key interfaces to complementary architecture frameworks if any (i.e., dedicated architecture framework),
• Definition and statement of work for customization & initialization of architecture support tools, including interfaces to complementary tools and repositories if any,
• Agreement on architecture principles applicable from vision to baseline. Agreed principles (or links to) are initialized within the architecture repositories at kick-off,
• Architecture management plan outline.</td></tr>
<tr><th>Stakeholders</th><th>Input Views</th><th>Output Views</th></tr>
<tr><td>• Architect, Project Manager,
• Representatives of plans, operations, legacy systems, standards, technology, regulations and laws),
• Specialists (security, safety, human factors, etc.),
• Sponsors.</td><td>L2, L3, L4, P3, P4, P8, Pr, A8.</td><td>A1 to 8 (i.e. meta-data, architecture plan and architecture summary documentation with references to input views.</td></tr>
</table>

Table 2-13 – Project: Establish Architecture Vision (AV)

Objectives	Task	
• Set project objectives from strategic goals, • Scope architecture sustaining business objectives: for target market, within key timeframes and milestones allowing the right effects /profits/savings and respecting local constraints & policies, • Define architecture outcome with regards to enterprise principles, • Identify architecture risks and define mitigation actions.	• Get stakeholder commitment on architecture work, • Validate stakeholder high level requirements, • Get stakeholders agreement on: enterprise motivation data setting and usage, architecture principles, architecture goals and drivers with regards to timeframes, • Analyse existing architecture baselines if any, • Validate architecture goals and drivers with regards to timeframes, • Identify interleaving with other projects with focus on critical milestones and interfaces, • Establish a statement of architecture work: initialize architectures comparison criteria, tailor the architecture development process (outcomes of each phase, synchronization, iterations and milestones), • Update architecture dashboard.	
Inputs	**Outputs**	
• Architecture management plan outline, • Request for architecture work including references to existing architecture baselines, • Committed architecture stakeholders, • (Identified) business goals and drivers, • Architecture principles, • Common architecture framework, • Initial Architecture dashboard.	• (Updated) architecture management plan, • Updated state of architecture work, • Preliminary architecture management plan, including architecture deliveries and reviews taking into account synchronization with related architecture projects, • Updated project motivation data (including top level requirements when necessary), • Initial architecture risk and mitigation plan, • Updated architecture dashboard.	
Stakeholders	**Input Views**	**Output Views**
• Architect, Project Manager, • Representatives of plans, operations, legacy systems, standards, and technology.	A1 to A8, Ar. C1 to C8, Cr. S1, Sr, L1, Lr, P1, Pr (of legacy).	A1 to 8 (i.e. meta-data, architecture plan and architecture summary documentation with references to input views.

<table>
<tr><th colspan="3">Table 2-14 – Project: Describe Alternatives of Architecture (AD)</th></tr>
<tr><th>Objectives</th><th colspan="2">Task</th></tr>
<tr><td>• Describe, starting from ‘as-is’ architecture and, in consistency with Enterprise Architecture principles, alternatives of solution architectures that meet project’s architecture vision.</td><td colspan="2">• Validate stakeholders’ key expectations and constraints,
• Confirm shared vision on architecture objectives, stakes, constraints and timeframes,
• Get agreement on projects architecture drivers in consistency with enterprise drivers,
• Describe identified architecture alternatives, using drivers to orient view selection and mappings,
• Review consistency of each alternative (i.e. described by a set of views) using audit matrixes,
• Update architecture dashboard.</td></tr>
<tr><th>Inputs</th><th colspan="2">Outputs</th></tr>
<tr><td>• Enterprise portfolios and reference architectures,
• Enterprise motivation data: including drivers,
• Project architecture motivation data: shared architecture vision, stakeholder’s needs / high level requirements and constraints and architecture drivers (DLODs, TEPIDOIL[21], PESTEL, and DOTMLPFI),
• Initialized architecture description framework and principles: selected description views, selected mapping views, traceability to customer requirements and max & minimum number of alternatives.</td><td colspan="2">Report on architecture description and findings:
• Identified and named architecture alternatives,
• Description of each alternative according to selected views and mappings,
• Gap analysis of each alternative with regards to expectations: milestone shift, capability metric, quality factor, technology maturity, etc,
• List of drivers used and justification for unused drivers,
• Updates architecture risk file and fall-back actions,
• Recommendations for trade-off and impacted drivers,
• Up to date architecture dashboard.</td></tr>
<tr><th>Stakeholders</th><th>Input Views</th><th>Output Views</th></tr>
<tr><td>• Architect, Project Manager,
• Representative of: plans, operations, legacy systems, standards, technology regulations and laws,
• Security architect, safety architect.</td><td>A1 to A8, Ar.
C8, Cr.
Si, Sr, Li, Lr, Pi, Pr (of legacy).</td><td>A1 to A8, Ar.
C1 to C8, Cr.</td></tr>
</table>

21 TEPIDOIL – Training, Equipment, Personnel, Information, Doctrine, Organization, Infrastructure, and Logistics.

<table>
<tr><th colspan="3">Table 2-15 – Project: Evaluate Alternatives of Architecture and Get Trade-Off (AE)</th></tr>
<tr><th>Objectives</th><th colspan="2">Task</th></tr>
<tr><td>• Compare identified alternatives of architecture and highlight key benefit of each, according to architecture drivers at both project and enterprise levels,
• Identify and report on the best candidate architecture with regards to needs and key assumptions,
• Identify sustainable trade-offs that:
- Reduce gaps to needs at a satisfactory level for stakeholders.
- Reduce sensitivity to possible changes.</td><td colspan="2">• Assess architecture consistency with regards to key (weighted) drivers and constraints and determine architecture gaps,
• Confirm/update architecture evaluation grid according to project motivation data,
• Confirm/update architecture goals and objectives,
• Conduct evaluation and comparison with regards to architecture goals and objectives,
• Determine trade-off proposals ensuring confirmed project objectives are met in consistency with enterprise constraints and principles,
• Get decisions from the architecture board (i.e. the board will have assessed trade-off proposal architectures with regards to key architecture drivers and constraints,
• Perform gap analysis (capability coverage, cost, availability, performance) on architecture trade-off,
• Update risk data and mitigation actions,
• Baseline trade-off architecture in the architecture repositories, including traceability links to rationale for evaluation and decision,
• Update architecture dashboard.</td></tr>
<tr><th>Inputs</th><th colspan="2">Outputs</th></tr>
<tr><td>• Statement of architecture work.
• Initialized evaluation and comparison grid,
• Weighted comparison criteria</td><td colspan="2">• Report on architecture evaluation activities:
• Score of assessed alternatives of architecture and identified trade-offs,
• Description of the trade-off, including key assumptions, concerned criteria and weights,
• Gap analysis: evaluates the distance of trade-off to architecture objectives (capability coverage, effectiveness, performances, cost, availability, risk),
• Updated high level Implementation requirements,
• Migration plan and migration strategy,
• Up to date architecture dashboard.</td></tr>
<tr><th>Stakeholders</th><th>Input Views</th><th>Output Views</th></tr>
<tr><td>• Architect, Project Manager,
• Representative of: plans, operations, legacy systems, standards, technology, regulations and laws,
• Security architect and safety architects,
• Representative of human factors
• Sponsor.</td><td>A4: evaluation method, evaluation criteria, objectives of trade-offs: what to optimize, why, when.
Ar: key milestones.
A8: constraints.
Views to compare
Cr: Expected and proposed.
L2-L3: expected and proposed.
P2: constraints and proposed.
A8: initialized and achievable by alternative.
S1: expected and achievable by alternative.</td><td>Updated A5/A6.
Compared views and value.
C2: actual phasing vs expected.
L2-L3: operational architecture effectiveness.
P2: impacts on Key interfaces and legacy, system qualities & performance.
A8: achievability of expected Technical Readiness Levels (TRLs).
S1: impact on expected quality and availability (migration, implementation and maintenance).</td></tr>
</table>

<table>
<tr><th colspan="3">Table 2-16 – Plan Migration (MP)</th></tr>
<tr><th>Objectives</th><th colspan="2">Task</th></tr>
<tr><td>• Coordinate various project impacted by the defined architecture,
• Elaborate implementation plan from a prioritized list of projects.</td><td colspan="2">• Analyse & confirm gap analysis with respect to architecture definition,
• Prioritize projects according to description of baseline:
- Estimate resources for migration using baseline of capability phasing, system evolution, system technology evolution, technology forecast.
- Perform cost/benefit analysis for each project.
- Identify high risk projects with respect to capability dependencies and projects' milestones,
• Generate a proposal migration roadmap,
• Establish a migration plan showing how existing systems will migrate to the architecture baseline,
• Identify impacts and issue change requests on baseline architecture.
- Architecture descriptions including phasing and mapping views
- Links to key drivers and constraints,
• List of standard products and required evolutions.</td></tr>
<tr><th>Inputs</th><th colspan="2">Outputs</th></tr>
<tr><td>• Baseline of architecture definition:
- Descriptions: capability, operational, system, technical, phasing, and mapping views.
- Traceability to architecture trade-off, hypotheses and rationale (motivation data).
- Traceability to top level requirements reflecting (and or having led to) architecture trade-offs (motivation data).
- Traceability to standard products/ building blocks (refer to project architecture repositories),
• Risk data & mitigation action list.</td><td colspan="2">• Impact analysis report,
• Detailed migration plan,
• If necessary, proposal to update architecture contract.</td></tr>
<tr><th>Stakeholders</th><th>Input Views</th><th>Output Views</th></tr>
<tr><td>• Architect, Project Manager,
• Representatives of plans, operations, legacy systems, standards, technology, regulations and laws,
• Security and safety architects,
• Representative of human factors,
• Sponsor.</td><td>A1 to A8, Ar.
C1 to C8, Cr.
S1 to S8, Sr.
L1 to L8, Lr.
P1 to P8, Pr.</td><td>A1 to A8, Ar.
C1 to C8, Cr.
S1 to S8, Sr.
L1 to L8, Lr.
P1 to P8, Pr.</td></tr>
</table>

Table 2-17 – Govern Application of Architecture (AG)		
Objectives	**Task**	
• Monitor application of architecture in multiple development & deployment projects, • Formulate recommendations and set a contract between architecture board and impacted projects.	• Monitor application of architecture in multiple development & deployment projects, • Formulate recommendations and set a contract between architecture board and impacted projects. For each impacted project: • Identify key architectural requirements, • Define conformance review plan and reviews according to the project's timeline, • Define and share conformance rules and criteria, • Perform architecture compliance reviews • Identified architectural gaps and formulate recommendations, • Document change requests to the baseline architecture.	
Inputs	**Outputs**	
• Architecture motivation data, • Request for architecture work, • Statement of architecture work, • Architecture vision, • Architecture repositories, • Architecture definition and associated change requests, including roadmap, transition scenario of each impacted projects and associated migration plans.	For each impacted project: • Status of projects' compliance to baseline architecture including impact analysis and identified gaps and recommendation to impacted projects, • Update to architecture state of work, • Update to project's architecture. • Compliance of developed and or deployed solution, • New change requests (if any) to in the architecture baseline.	
Stakeholders	**Input Views**	**Output Views**
• Architect, Project Manager, • Representatives of plans, operations, legacy systems, standard, technology watch, regulations and laws), • Specialists (security, safety, human factors, etc.), • Sponsors.	A1 to A8, Ar. C1 to C8, Cr. S1 to S8, Sr. L1 to L8, Lr. P1 to P8, Pr.	A1 to A8, Ar. C1 to C8, Cr. S1 to S8, Sr. L1 to L8, Lr. P1 to P8, Pr.

<table>
<tr><th colspan="3">Table 2-18 – Project: Decide on Architecture Changes (AC)</th></tr>
<tr><th>Objectives</th><th colspan="2">Task</th></tr>
<tr><td>• Ensure that changes to the architecture are decided and managed in a controlled manner,
• Establish an architecture change management process for the new architecture that will be used along governance of implementation & deployment projects.</td><td colspan="2">• Tailor architecture change management process,
• Collect and classify architecture change requests,
• Develop change requirements to meet architecture goals as defined in the vision.
• Define the nature and impact of change and get agreements from the architecture board,
• Manage risks.</td></tr>
<tr><th>Inputs</th><th colspan="2">Outputs</th></tr>
<tr><td>• Request for architecture work identified at trade-off analysis and decision,
• Statement of Architecture work,
• Architecture vision,
• Architecture repositories,
• Architecture definition document and roadmap,
• Motivation data:
  • Change requests due to changes identified in enterprise business, technology or standards,
  • Transition scenario,
  • Architecture state of work,
  • Implementation and migration plan, security, safety, maintainability, operational costs, human comfort, configurability,
  • Evolution of enterprise and business context since the last architecture change,
  • Up to date opportunity reports.
  • Up to date Technology maturity status report.</td><td colspan="2">• Agreement for architecture changes,
• Architecture updates,
• New request for architecture work (to initiate a new cycle of the method)
• Updated Statement of architecture work,
• Updated architecture statement of work,
• Notification of changes toward architecture stakeholders.</td></tr>
<tr><th>Stakeholders</th><th>Input Views</th><th>Output Views</th></tr>
<tr><td>• Architect, Project Manager,
• Representatives of plans, operations, legacy systems, standard, technology, regulations and laws),
• Specialists (security, safety, human factors, etc.),
• Sponsors. All existing views. All Views and perspectives impacted by architecture change (capabilities, system, capability increment milestones, functions, services, organization, activities, etc.), accepted changes and impacts.</td><td>All existing views.</td><td>All views and perspectives impacted by architecture change (capabilities, system, capability increment milestones, functions, services, organization, activities, etc.), accepted changes and impacts. A1-8 and/or C1-8 and S1-8 or and /or L1-8, P1-8, Ar, Cr, Sr, Lr, Pr.</td></tr>
</table>

Table 2-19 – Project: Manage Architecture Motivation Data & Dashboard (MD)	
Purpose	**Tasks**
• Set and maintain architecture up-to-date motivation data, • Monitor architecture progress and stop activities according to enterprise policy and stakeholder's expectations.	• Initialize motivation data starting from project landscape. • Check consistency of architecture principles with enterprise directives, • Check consistency of constraints with enterprise directives: economic (cost, value, risk), missions, physical (weather, electromagnetism compatibility, terrain, human factors, security and safety, export and regulation, skills, • Identify the effective drivers of architecting activities: choose DLOD, PESTEL, DOTMPLFI, etc. according to the analysis of stakeholders needs, • Check the joint impact of pre-cited factors, on the current baseline, whether implemented or on the way to be. Hint: the impact may be described using NAFviews, to be completed by top level customer, user or technology related requirements, • Set principles for architecture change decision, • Revisit motivation data according to outputs of the last iteration of the vision stage, in terms of: evolution of contexts and needs, evolution of norms, standards and regulations, release, update or obsolescence of domain, technology, business, political, and societal conditions, changes to doctrine, business, technology and enterprise strategy, • Agree on priority over expected capabilities from business viewpoint, • Use capability dependency and capability phasing to highlight critical milestones, • Agree on weight of criteria selected to evaluate and alternatives of architecture, • Revisit stakeholder requirements according to priority and weighted criteria, • Initialize architecture dashboard with agreed data (weights, dependencies, priorities, criteria, objectives, roadmaps), • Log the context of architecture assessment and trade-offs at each decision point, • Update dashboard, check and manage alerts, • Mark selected/discarded/changed artefacts at each decision point, • Trace towards inputs and document rationale of each decision.
Inputs	**Outputs**
• Architecture management plan • Elements from project architecture landscape, • Elements of Architecture vision: planning of architecting phases, initial milestones for synchronization with enterprise, initial milestones for synchronization between project phases.	• Architecture management plan update, • Dashboard featuring: key architecture milestones: Phase milestone, synchronization milestones (inter-phases, enterprise to project), stop criteria, progress of each phase of architecting vs. project milestones, alert icons (on phases, synchronization between phases and/or with enterprise milestone), decision points, marked artefacts, • Vision models and documentation published in the architecture repositories.

Stakeholders	Input views	Output views
• Architect, Project Manager, • Representatives of plans, operations, legacy systems, standard, technology watch, regulations and laws, • Specialists (security, safety, human factors, etc.) • Managers of implementation projects. • Sponsors.	C1 to C3, Cr, Lr, A1 to A8.	Updates of C1 to C3, A1 to A8, Ar, Cr, Sr, Lr, Pr.

9 FOUNDATION FOR ARCHITECTING

This section describes the common methodological elements necessary to elaborate either Enterprise or Project Architecture Frameworks.
These elements are related to activities and architecture data:

- Architecture Principles,
- Architecture Capabilities,
- Architecture Patterns,
- Architecture Assets, and
- Organization for Architecting.

9.1 Architecting Principles (Foundation for Best Practices)

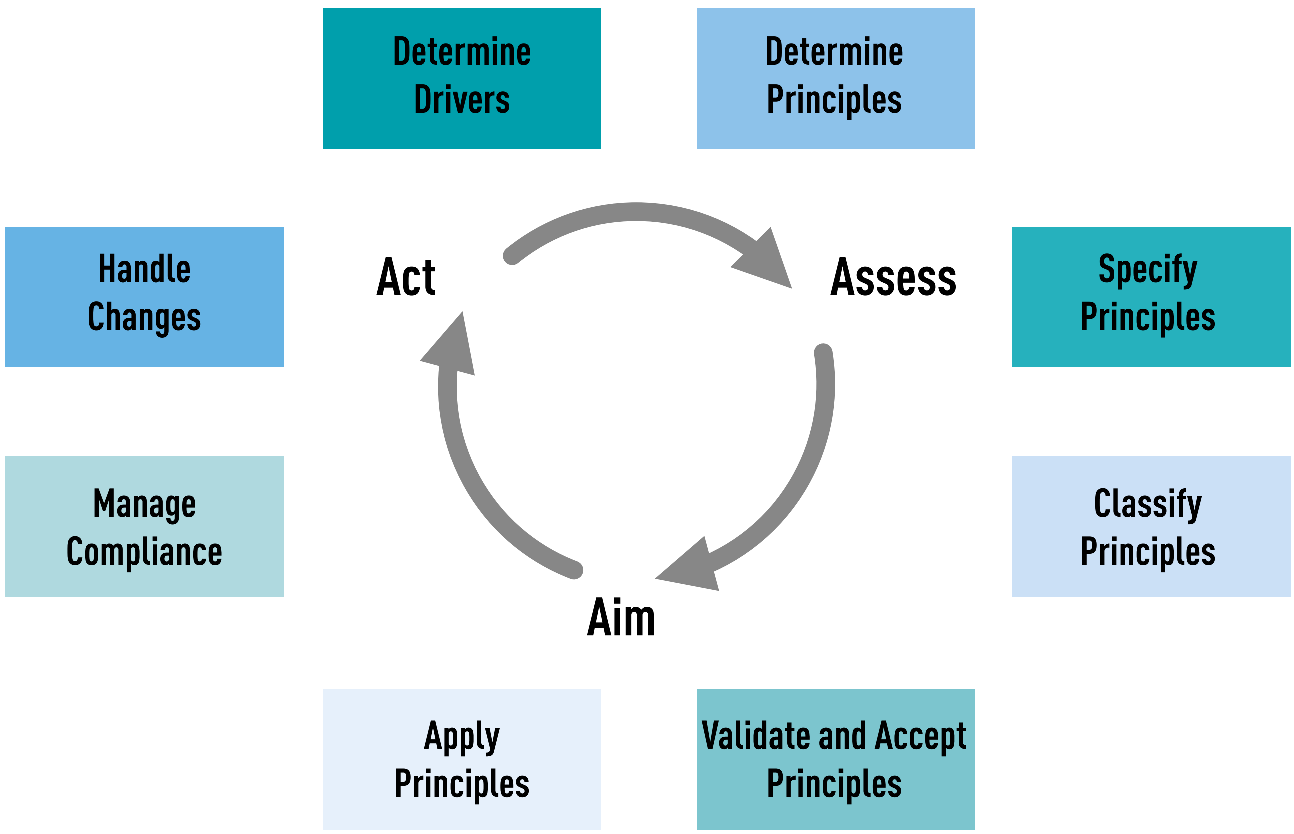
The approach described in this section for establishing architecture principles is significantly based on the book written by Danny Greefhorst and Erik Proper Greefhorst, 2011.

9.1.1 Overview

Figure 2.15 describes:

- the process starting with the determination of the drivers, which are the foundation for architecture principles,
- in subsequent sub-processes the architecture principles themselves are determined, specified, classified, validated, and applied,
- the next sub-process is using architecture principles to determine whether activities comply with the architecture, and
- the final sub-process intends to handle changes to the architecture, which may restart the initial sub-process.

Figure 2-15: Architecture Principles Definition and Management Activities



9.1.2 Definitions for Architecture Principles

Normative	A declarative statement that normatively prescribes a property of something.
Credo	A normative principle expressing a fundamental belief.
Design	A normative principle on the design of an artefact. As such, it is a declarative statement that normatively restricts design freedom.
Architecture	A normative principle on the orientation towards an effective artefact.

9.1.3 Description of Sub-Processes

- Define drivers where the relevant inputs for determining architecture principles are collected from the enterprise and project motivation data, such as the goals and objectives, opportunities, issues and risks.
  - Drivers are ideally defined outside the scope of the architecture activities (ideally need to be gathered explicitly before architecture principles can be identified).
  - Drivers that are not explicitly documented may have to be elicited from stakeholders.
  - Architects have to ensure that the definitions of these drivers are current, and to clarify any areas of ambiguity.
  - The exact nature of the goals depends on the exact scope and context of the architecture engagement.
  - The goals and issues are the basic drivers that should be addressed. Others may be added in later iterations.
  - Having identified the types of drivers, the next step is to determine which information on these drivers is needed in order to determine the architecture principles.
  - Validate the drivers with the stakeholders (What may seem a driver for one stakeholder, may seem irrelevant for someone else).
  - The final step in the determination of drivers is their explicit specification in the form of an architectural requirement. This results in a list of statements with a unique identification that is the basis for the determination of architecture principles. It thereby enables traceability from drivers to architecture principles, as well as requirements management of these drivers,
- Determine principles where the drivers are translated to a list of (candidate) architecture principles. At this stage the architecture principles can be considered Credos.
  - Generate candidate principles: generates a list of candidate architecture principles that address the drivers.
  - Select relevant principles: selects those architecture principles that are relevant to the specific architectures.
  - Formulate principle statements: specializes or generalizes the candidate architecture principle statements into the proper abstraction level,
- Specify principles where the candidate principles are specified in detail, including their rationale and implications. This sub-process translates architecture principles from Credos to Norms.
  - After the architecture principles have been determined they need to be specified in more detail. Further detailing of the architecture principle is a prerequisite for actually using it to restrict design freedom,
- Classify principles where architecture principles are classified in a number of dimensions to increase their accessibility.
  - After the architecture principles have been specified it is useful to classify them along the dimensions that were described in the previous sub-process to ease their accessibility and maintainability.
  - The dimensions proposed are type of information, scope, genericity, details level, stakeholder, transformation, quality attribute, meta-level and representation,
- Validate and accept principles where architecture principles, their specifications and classifications are validated with relevant stakeholders and formally accepted.
  - Quality criteria that can be used to determine the quality of the architecture principles. The quality criteria generally proposed are: specific, measurable, achievable, relevant and time framed. For sets of architecture principles the quality criteria are: representative, accessible

and consistent. The review process as well as the criteria should, however, be customized and refined to the organizational context,

- Apply principles where architecture principles are applied to construct models and derive decisions in downstream architectures, requirements and applications.
  - Using architecture principles requires a good understanding on the artefacts that are impacted by them,
- Manage compliance where architects ensure that the architecture principles are applied properly, and dispensations for deviations may be given. Every architecture principle can be scored on the scale described in the following table.

Table 2-20 – Level of Compliance

Level of Compliance	Description
Not Conformant	Some part of the specification of the artefact is not in accordance with the architecture principle.
Potentially Compliant	There is not enough specified in the artefact in order to determine whether it is in accordance wwith the architecture principle.
Compliant	Everything specified in the artefact is in accordance with the architecture principle, but some relevant implications of the architecture principle are missing in the artefact.
Potentially Conformant	Everything specified in the artefact is in accordance with the architecture principle, but there is not enough specified in order to determine that all relevant implications of the architecture principle are embedded in the artefact.
Fully Conformant	Everything specified in the artefact is in accordance with the architecture principle, and all relevant implications of the architecture principle are embedded in the artefact.

- Handle changes where the impact of all sorts of changes on the architecture principles is determined and new method iterations may be initiated.

A change management process is needed to guide the organization in handling all these drivers for change. The most important part of such a process is a classification scheme of types of changes that provides guidance on the appropriate steps to take.
Also, there should be a standard periodic architecture refreshment cycle in which changes can be incorporated. See the "Decide on architecture change stage of the NAVv4.

9.1.4 Architecture Principles in NAFv4

- The Architecture activities for both enterprise and projects are grouped in 8 stages. These stages are all concerned with architecture principles.
  - They are the first architecture principles to be applied in the stage dealing with establishment of the architecture landscape (AL), reviewed and extended in the architecture vision (AV) and checked during the architecture description and evaluation stages (AD & AE).
  - Changes to them are handled during the stage dealing with the decisions on the architecture change (AC).
  - The establishment of the architecture landscape builds the foundation for the architecture and is where the main architecture principles are described.
  - Architecture principles are positioned as derivatives of enterprise principles, which should be defined outside the architecture processes.
  - However, depending on how such principles are defined and promulgated within the enterprise, it may be possible for the set of architecture principles to also restate, or cross-refer to a set of enterprise principles, enterprise goals, and strategic enterprise drivers defined elsewhere within the enterprise,

- These principles are derived and adapted for the architecture activities in the projects according to the architecture motivation data in these projects,
- The architect normally needs to ensure that the definitions of these enterprise and project principles, goals and strategic drivers are current, and to clarify any areas of ambiguity,
- The architecture principles are identified and established after the organizational context is understood and a tailored architecture framework is in place in the enterprise and in the projects,
- Architecture principles should have a name, statement, rationale and implications,
- The architecture description and architecture evaluation stages can work on separate Viewpoints for definition and evaluation of Views according to stakeholder concerns. For example:
  - Operational Views.
  - System Views.
  - Technical Views,
- Architecture activities will use the architecture principles that were defined and maintained during the establishment of the architecture landscape and architecture vision elaboration to build the specific architecture domains upon,
- Also, it may work upon architecture principles that are specific to the architecture perspectives like: business architecture principles and data architecture principles.

9.1.5 Architecture Capabilities

- Architecture capabilities comprise any necessary resource, capacity and ability necessary to perform architecture activities at Enterprise or project level:
  - human capabilities: the ability to perform roles and manage responsibilities, as of disciplines and specialties, with the right skills & competencies, and
  - technical capabilities: the ability to support human capabilities and automate partly of entirely their activities and outcomes (ex. tooling capabilities),
- A capability life cycle spans needs, requirements, acquisition, in-service and disposal phases. A capability has attributes and measure of effectiveness (e.g. effect, scale, time) and is defined independently from implementation means, and
- Architecture capabilities are used in various combinations to achieve outcomes. A capability is usually described as one or more sequences of activities (called operational threads). The ability to execute an activity depends on many factors identified at landscape establishment and enriched throughout architecture stages.

9.1.6 Recommended Patterns for Architecture and Architecting

- An architecture pattern records decisions taken by many architects in many projects and organizations over many years in order to answer to a recurring architecture questions through different drivers and involving multiple concerns,
- An architecture pattern is a reusable description of an architecture view as described in the NAFv4 grid. The problem to solve may concern a roadmap, the modes and states of a system, a recurring a course of operations in a well-known operational domain. Therefore, a multi-viewpoints problem may need many patterns in combination to meet architecture objectives, and
- Patterns are managed as assets: they are documented in reference libraries and may be found classified in catalogues. They have an owner and are subject to approval by a board of architects.

9.1.7 Architecture Assets

- Architecture assets are any architecture element that can be considered in the Enterprise. These assets are either used at enterprise level or shared between projects,
- The architecture assets basically include deliverable and building blocks. Architecture patterns can also be considered as assets to some extent. However assets cannot structured without consideration of:
  - requirements, architecture training courses, architecture training facilities,
  - viewpoints, models, views, diagrams, patterns and other artefacts,
  - catalogues (synonyms are portfolios and libraries) of: patterns, architecture projects, architecture views and main architecture documents for instance, and

  - associated baselines: reference requirements baseline, patterns baseline, architecture model and views baseline, architecture project catalogue baseline.
- A real ontology is needed here to describe formally the Architecture Data,
- Some examples at this point are:
  - a set of services exposes a Catalogue of Services, and
  - an Architecture View considered as a Solution Building Block.
- A diagram considered as a Requirement (i.e. an expectation).

Chapter 3 - Viewpoints

1 INTRODUCTION

1.1 Architecture Descriptions

1.1.1 Architecture Descriptions typically address a set of related concerns and is tailored for specific stakeholders. Views are an ideal mechanism to purposefully convey information about specific concerns. A View is specified by means of a Viewpoint, which prescribes the concepts, models, analysis techniques, and visualizations that are provided by the View.

A View is what you see.
A Viewpoint is where you are looking from.

1.1.2 ISO/IEC 42010 provides the following definitions relevant to this chapter:

Term	Meaning
Architecture Description (AD)	Work product used to express an architecture.
Architecture View	Work product expressing the architecture of a system from the perspective of specific system concerns
Architecture Viewpoint	Work product establishing the conventions for the construction, interpretation and use of Architecture Views to frame specific system concerns.
(System) Concern	Interest in a system relevant to one or more of its stakeholders. 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.
(System) Stakeholder	Individual, team, organization, or classes thereof, having an interest in a system.
Model Kind	Conventions for a type of modelling. Examples of Model Kinds include data flow diagrams, class diagrams, Petri nets, balance sheets, organization charts and state transition models.

1.1.3 An Architecture Description includes one or more Architecture Views. An Architecture View (or simply a View) addresses one or more of the concerns of a stakeholder for the system of interest.

1.1.4 A View expresses the architecture of the system of interest in accordance with an Architecture Viewpoint (or simply a Viewpoint).

1.1.5 A Viewpoint frames one or more concerns. A concern can be framed by more than one Viewpoint. A Viewpoint establishes the conventions for defining and evaluating Views to address concerns framed by that Viewpoint. Viewpoint conventions can include languages, notations, Model Kinds, design rules and/or modelling methods, and other operations on Views.

Viewpoints are a means to focus on particular subjects and aspects of stakeholder concerns.

1.1.6 The NATO Architecture Framework (NAF) provides a set of standardized Viewpoints that can be used for NAF-Compliant architecture efforts. However, not all of the standardized Viewpoints will be required for each architecture effort, and for specific architecture efforts additional Viewpoints might be suitable.

NAF neither mandates the use of all standardized Viewpoints, nor does it exclude the usage of additional Viewpoints, if required, to address stakeholder concerns.

2 NAF GRID REPRESENTATION

2.1 Description

2.1.1 The NAF Grid Representation (see Figure 3-1 below) is a two-dimensional classification scheme for the standardized NAFviewpoints, which serve as the baseline for any NAF-Compliant architecture effort. However, the selection of Viewpoints must be tailored to the specific architecture effort, i.e. suitable Viewpoints need to be identified in the grid, and additional Viewpoints must be defined, if and when required.

Figure 3-1 - NAF Grid Representation

					Behaviour					
	Taxonomy	Structure		Connectivity	Processes	States	Sequences	Information	Constraints	Roadmap
Concepts	C1 Capability Taxonomy NAV-2, NCV-2	C2 Enterprise Vision NCV-1		C3 Capability Dependencies NCV-4	C4 Standard Processes NCV-6	C5 Effects NOV-6b		C7 Performance Parameters NCV-1	C8 Planning Assumptions	Cr Capability Roadmap NCV-3
	C1-S1 (NSOV-3)									
Service Specifications	S1 Service Taxonomy NAV-2, NSOV-1			S3 Service Interfaces NSOV-2	S4 Service Functions NSOV-3	S5 Service States NSOV-4b	S6 Service Interactions NSOV-4c	S7 Service I/F Parameters NSOV-2	S8 Service Policy NSOV-4a	Sr Service Roadmap
Logical Specifications	L1 Node Types NAV-2	L2 Logical Scenario NOV-2	L2-L3 (NOV-1)	L3 Node Interactions NOV-2, NOV-3	L4 Logical Activities NOV-5	L5 Logical States NOV-6b	L6 Logical Sequence NOV-6c	L7 Logical Data Model NSV-11a	L8 Logical Constraints NOV-6a	Lr Lines of Development NPV-2
					L4-P4 (NSV-5)					
Physical Resource Specifications	P1 Resource Types NAV-2, NSV-2a,7,9,12	P2 Resource Structure NOV-4,NSV-1		P3 Resource Connectivity NSV-2, NSV-6	P4 Resource Functions NSV-4	P5 Resource States NSV-10b	P6 Resource Sequence NSV-10c	P7 Physical Data Model NSV-11b	P8 Resource Constraints NSV-10a	Pr Configuration Management NSV-8
Architecture Meta-Data	A1 Meta-Data Definitions NAV-3	A2 Architecture Products		A3 Architecture Correspondence ISO42010	A4 Methodology Used NAF Ch2	A5 Architecture Status NAV-1	A6 Architecture Versions NAV-1	A7 Architecture Meta-Data NAV-1/3	A8 Standards NTV-1/2	Ar Architecture Roadmap

2.1.2 The grid approach presents the NAFviewpoints by Subject of Concern (rows) and by Aspect of Concern (columns). The NAF is arranged as a grid with columns as set of broad Model Kinds represented in Table 3.1.

Table 3-1 – Description of the Columns in the Grid

Aspects	Description
Taxonomy	**Specialization hierarchies of architecture elements such as capabilities, services, etc.**
Structure	**How elements are assembled (enterprises, nodes, resources, etc.).**
Connectivity	Everything from high-level capability dependencies to detailed system connectivity.
Behaviour	How things work: • Processes - Process flows and decomposition. • States - Allowable state transitions. • Sequences - How things interact and in what order.
Information	**What information/data is used, and how it is structured.**
Constraints	**Rules that govern the enterprise, nodes, resources, etc.**
Roadmap	Project timelines and milestones affecting the elements in the architecture.

2.1.3 The NAFviewpoints retain an equivalence with the NAFv3 Views[22], albeit with names that better describe their purpose, as indicated in Table 3-1: Mapping of NAFv3 Views to NAFv4 Viewpoints:

Table 3-2: Mapping of NAFv3 Views to NAFv4 Viewpoints

NAFv3 View	NAFv4 Viewpoints
Capability (NCV)	Concepts (C)
Service-Oriented (NSOV)	Service (S)
Operational (NOV)	Logical (L)
Systems (NSV)	Physical Resource (P)
All Views (NAV)	Architecture Meta-Data (A)

2.1.4 Each cell at the intersection of the rows and columns is a Viewpoint (usually an existing NAFv3 View). The new approach is Information-Centric. It divides the framework up into categories of architectural information rather than how the information is presented.

2.1.5 Most of the NAFv3 Views match one cell (Viewpoint). However, because the grid is based on the type of information, rather than how it is presented, there are cases where a cell covers more than one NAFv3 View (usually this is where there is a graphical View and a tabular one showing the same information). There are also cases, there are no corresponding Views. Most of these are left blank on the grid, recognizing there is no current requirement in the NAF for this information. There are two cells (C5, Effects, and Sr, Service Roadmap) where there is meta-model coverage, but no equivalent View in the NAFv3 specifications. Some NAFv3 Views are not included – notably the Technology and Standards Forecasts, and the NAV-1 (Overview and Summary Information) as these are covered in more detail in the Architecture Meta-Data Viewpoints.

2.1.6 In order to deal with concepts such as actual organizations and fielded capabilities, the NAF grid approach moves these to the physical Viewpoints. Finally, some NAFv3 Views existed only to document the mapping between other Views. These are shown as interstitial Viewpoints (C1-S1, Capability to Service Mapping, and L4-P4, Activity to Function Mapping) in the grid.

2.1.7 The remainder of this document provides for each Viewpoint the following information:

- **Name and Description of the Viewpoint**, and an indication of mandatory and optional information that is to be provided by corresponding Views,
- **Concerns Addressed**, to identify the examples of stakeholder concerns addressed by the View,
- **Usage**, providing examples of use cases for Views of this Viewpoint,
- **Representation**, providing examples of Model Kinds that can be used to represent Views of this Viewpoint. These Model Kinds are not mandatory, and other Model Kinds can be used as well if more suitable for the intended purpose, and
- **Example**, providing an illustrative example for the View. The examples do not imply any mandate to use a specific Model Kind, or notation.

22 Post NAFv3, the term 'View' is used to refer to a populated View within a particular architecture, in accordance with ISO/IEC/IEEE 42010.

3 CONCEPT VIEWPOINTS

Table 3-3 – Concept Viewpoints

				Behaviour					
	Taxonomy	Structure	Connectivity	Processes	States	Sequences	Information	Constraints	Roadmap
Concepts	C1 Capability Taxonomy NAV-2, NCV-2	C2 Enterprise Vision NCV-1	C3 Capability Dependencies NCV-4	C4 Standard Processes NCV-6	C5 Effects NOV-6b		C7 Performance Parameters NCV-1	C8 Planning Assumptions	Cr Capability Roadmap NCV-3

The Viewpoints in the Concepts row of the NAF grid support the process of analyzing and optimizing the delivery of capabilities in line with the enterprise's strategic intent. This is achieved by capturing the enterprise's strategic vision and concepts and capabilities (C2 Viewpoint). These capabilities can be organized into a taxonomy (C1 Viewpoint) and then augmented with schedule data (Cr Viewpoint) and measures of effectiveness (C7 Viewpoint). In addition, dependencies between capabilities (C3 Viewpoint) can be captured, enabling capability options to be built in a more coherent manner, and effective trade-offs to be conducted (e.g. across common funded programmes).

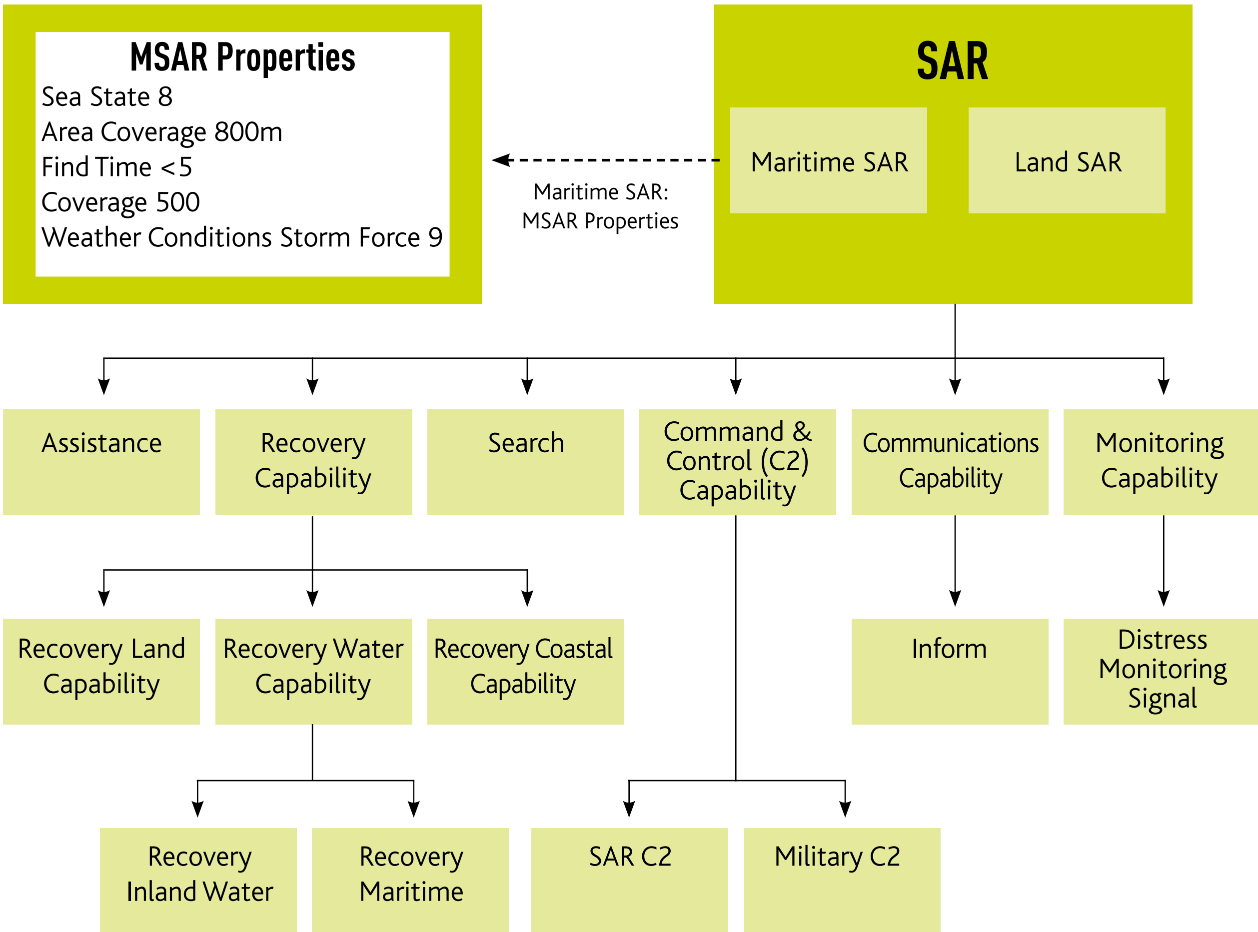
In the NAF, a capability is a description of an ability to do something.

3.1 C1 – Capability Taxonomy	NAFv3: NCV-2
The C1 Viewpoint is concerned with the identification of capabilities, and their organization into specialization hierarchies (taxonomies) independent of their implementation and may be referenced in whole or part by, or used in, describing multiple architectures (e.g. a C1 View at Enterprise-level will be referenced by C1 Views at the Capability-level). Views implementing this Viewpoint • Shall include all capabilities relevant for the architecture. • Shall organize all capabilities into a specialization hierarchy. • May include Measures of Effectiveness (MoE).	
CONCERNS ADDRESSED	**USAGE**
• Capability Planning. • Capability Management.	• Identification of existing and required capabilities. • Source for the derivation of cohesive sets of Key User Requirements (KURs). • Providing reference capabilities for multiple architectures.
REPRESENTATION	
• Tabulation. • Hierarchical (Connected Shapes). Class Diagram (with generalization relationships and property definitions).	

EXAMPLE

The following example uses a hierarchical diagram to depict the individual capabilities and their place in the taxonomy.

Figure 3-2: Example C1 View



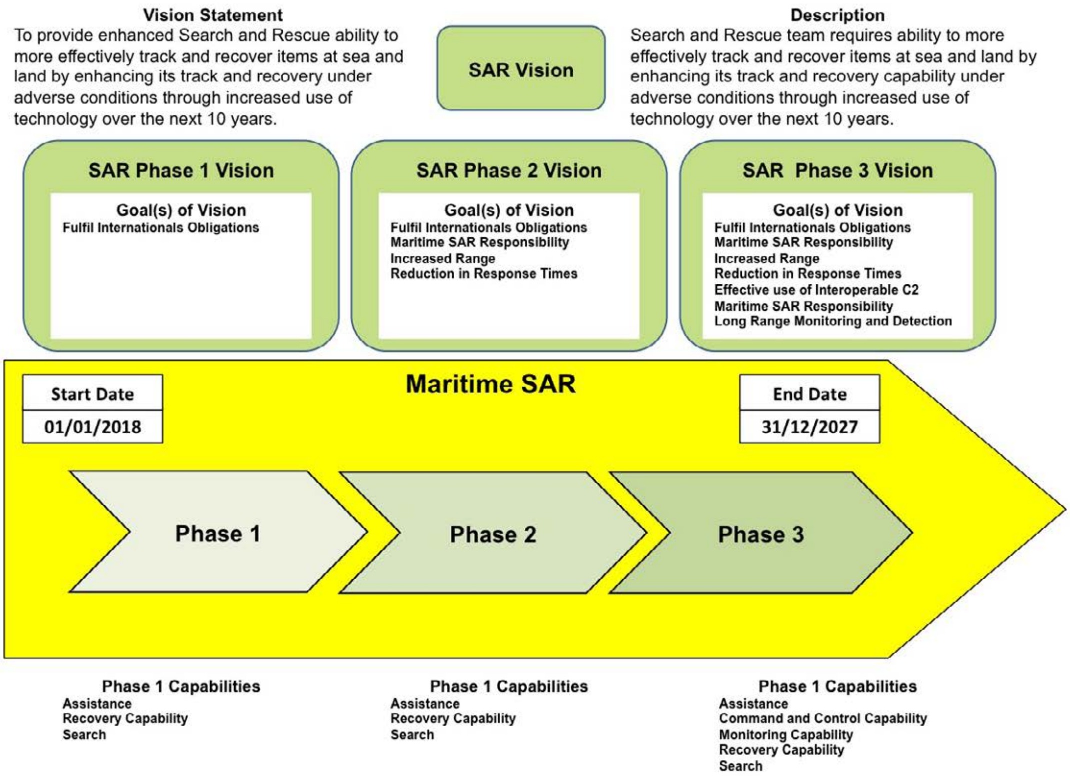
The capabilities in a C1 View are related by specialization relationships that assert one capability is a special case of another (e.g. Recovery Capability is specialized into Recovery Land Capability, Recovery Water Capability and Recovery Costal Capability in above example).

3.2 C2 – Enterprise Vision	NAFv3: NCV-1
The The C2 Viewpoint is concerned with scoping the architecture effort and providing the strategic context for the capabilities described in the architecture. Views implementing this Viewpoint: • Shall describe the vision and goals for the capabilities in scope for a defined period (or periods) of time. • May include desired outcomes and measurable benefits associated with the goals. • May link the capabilities to enduring tasks.	
CONCERNS ADDRESSED	**USAGE**
• Enterprise Strategy. • Capability Planning.	• Capture and communication of the strategic vision related to capability evolution. • Identify the capabilities required to meet the vision and goals. • Identify the required timescales for the capabilities (as opposed to Cr which provides a summary of when projects are estimated to deliver capability). • Identify any enduring tasks the enterprise performs. • Provision of a blueprint for a transformational initiative.
REPRESENTATION	
• Structured Text. • UML Composite Structure Diagram. • SysML Structural Diagrams.	

EXAMPLE

This is an example of a C2 View that shows the vision statement, three phases to achieve this vision, goals for each phase, and linkage to capabilities exhibited by each phase.

Figure 3-3: Example C2 View



3.3 C3 – Capability Dependencies

NAFv3: NCV-4

The C3 Viewpoint is concerned with identification of dependencies between capabilities, and defining the logical composition of capabilities (i.e. capability clusters).

Views implementing this Viewpoint:

- Shall include all dependencies between capabilities relevant for the architecture.
- May defines logical groupings of capabilities by means of composition.
- May include capability specializations (Note, this can also be expressed in a C1 View).

CONCERNS ADDRESSED	USAGE
• Capability Management.	• Analysis of dependencies between capabilities and between capability clusters. • Impact analysis for capability options, disposal of capabilities. • Highlight potential integration requirements and the interactions needed between acquisition projects in order to achieve the overall capability.

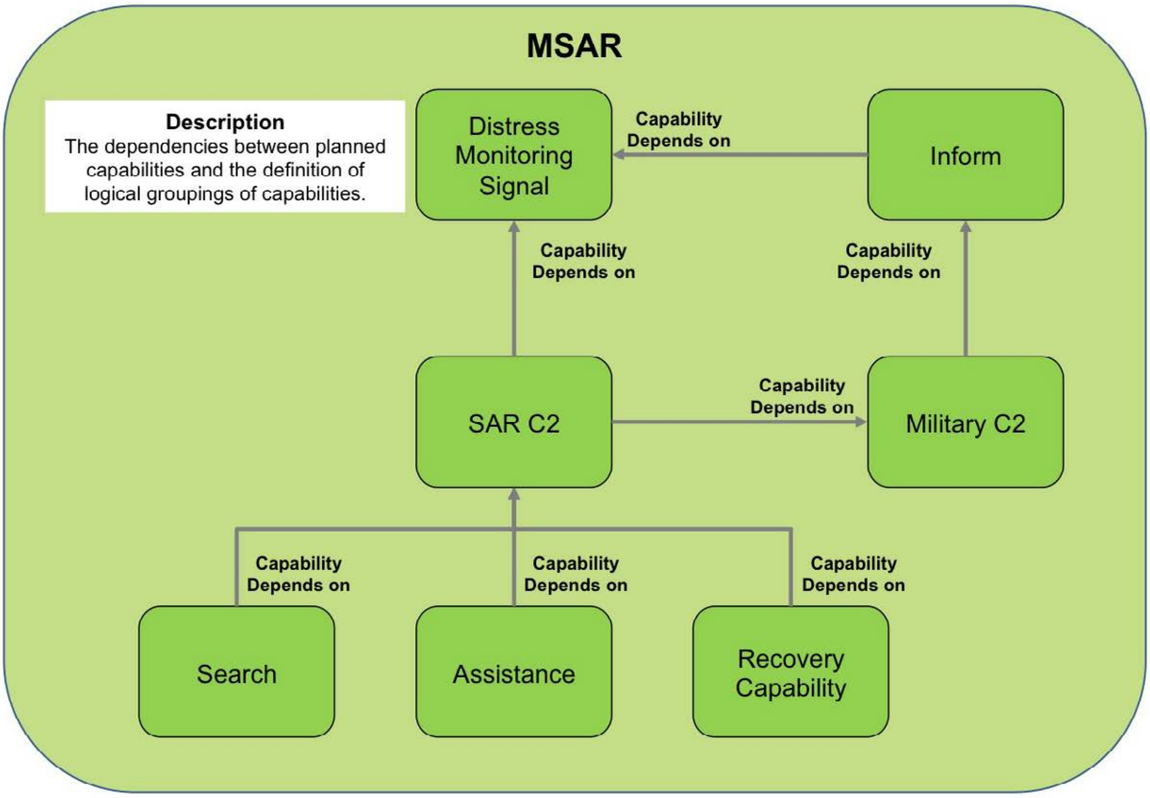
REPRESENTATION

- ‘Nested box’ diagram.
- UML Class diagram.
- UML Composite Structure diagram.
- SysML Structural diagram.

EXAMPLE

Figure 3-4 below shows a dependency diagram showing capabilities and dependencies between capabilities. A diagram like this can support the appropriate sequencing of capability development, e.g. before a SAR C2 capability gets implemented a Distress Monitoring capability needs to be in place.

Figure 3-4: Example C3 View



3.4 C4 – Standard Processes	NAFv3: NCV-6
The C3 Viewpoint is concerned with identification of standard activities (e.g. doctrinal) and optionally with their traceability to the capabilities supporting them. Views implementing this Viewpoint: • Shall identify all standard activities relevant for the architecture. • May provide a composition of these standard activities. • May link standard activities to supporting capabilities. A standard process list, in whole or parts, may be referenced by, or used in describing, multiple architectures (e.g. a C4 View at enterprise-level will be referenced by C4 Views at the capability-level).	
CONCERNS ADDRESSED	**USAGE**
• Doctrine Production. • Operational Analysis.	• Specification of doctrine. • Tracing capabilities to enduring tasks. • Tracing capabilities to standard operational activities. • Capability audit.
REPRESENTATION	
• Tabular. • Tracing Diagram.	

EXAMPLE

A C4 is usually shown in the form of a table, optionally listing the supported capabilities and enduring tasks, as shown in Figure 3-5.

Figure 3-5: Example C4 View

	Capability Maps to Activity	Capability has sub-components
MSAR		
Assistance	Locate Victim Monitor Health	
Command & Control Capability		Military C2 SAR C2
Military C2		
SAR C2	Coordinate Search Plan Search	
Communications Capability		Inform
Inform		
Monitoring Capability		Distress Monitoring Signal
Distress Monitoring Signal	Analyse Distress Signal Send Warning Order	
Recovery Capability	Maritime Rescue Provide Medical Assistance Recover Victim	Recovery Coastal Capability Recovery Land Capability Recovery Water Capability
Recovery Coastal Capability		
Recovery Land Capability		
Recovery Water Capability		Recovery Inland Water Recovery Maritime
Recovery Inland Water		
Recovery Maritime		
Recovery Maritime Inclement Weather		
Recovery Maritime Normal		
Search	Coordinate Search Execute Search Plan Search	

3.5 C5 – Effects

NAFv3: NOV-6B

The C5 Viewpoint is concerned with identifying and describing effects of capabilities.
Views implementing this Viewpoint:

- Shall define effects relevant for the architecture effort.
- Shall assign effects to capabilities.
- May identify start and/or end dates of effects.
- May identify resource types associated to start and end dates of effect.
- May show a specialization hierarchy of effects.

CONCERNS ADDRESSED	USAGE
• Operational Analysis. • Analysis of non-functional properties.	• Characterization of the expected results capabilities, positive or negative. • Analysis of cumulative effects. • Analysis of persistence of the effects. • Tracing the operational states and modes with regards to the effects.

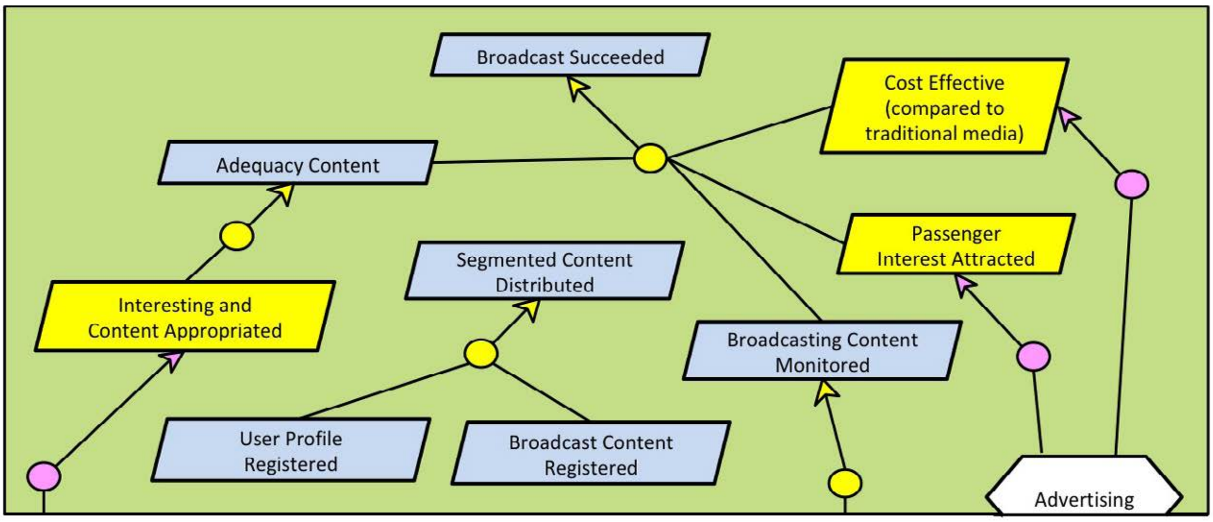
REPRESENTATION

- Tabular.
- Structural diagram.
- Histogram.
- Finite state diagram.

EXAMPLE

Example below shows a graphical depiction of effects and their relationships.

Figure 3-6: Example C5 View



3.6 C6 – Not Used

3.7 C-7– Performance Parameters

NAFv3: NCV-1

The C7 Viewpoint is concerned with the identification and description of measure categories, and identification of capabilities to which they are applicable.

Views implementing this Viewpoint:

- Shall identify all measure categories relevant for the architecture.
- May link measure categories to capabilities.

CONCERNS ADDRESSED

- Capability Planning.
- Capability Management.
- User Requirement Specification.

USAGE

- Setting Capability Requirements.
- Military Estimates.
- Strategic Reviews.
- Planning Assumptions.

REPRESENTATION

- Tabular (capabilities on one axis, measure categories on the other).
- UML Classes with property definitions.

EXAMPLE

C7 presents measure categories that can be used to judge the level of capability. It traces measure categories to the capabilities for which they are relevant.

Figure 3-7: Example C7 View

Enterprise Vision Applies to Phase							
SAR Performance Parameters	**Description**	**Enterprise Version Applied to Phase**	**Enterprise Phase Capabilities**	**Element as Performance Measures**	**State**	**Actual**	**Goal**
SAR Vision	Search and Rescue team requires ability to more effectively track and recover items at sea and land by enhancing its track and recovery capability under adverse conditions through increased use of technology over the next 10 years.	Phase 1	Assistance	Assistance Range from Station	Green	180	
				Assistance Response Time	Amber	8	
				Assistance Weather Conditions	Amber	3	
			Recovery Capability	Recovery Capability Range from Station	Amber	150	
				Recovery Capability Response Time	Red	9	
				Recovery Capability Weather Conditions	Amber	4	
			Search	Search Range from Station	Red	300	
				Search Response Time	Red	8	
				Search Weather Conditions	Green	4	
		Phase 2	Assistance	Assistance Range from Station	Green	180	
				Assistance Response Time	Amber	8	
				Assistance Weather Conditions	Amber	3	
			Recovery Capability	Recovery Capability Range from Station	Amber	150	
				Recovery Capability Response Time	Red	9	
				Recovery Capability Weather Conditions	Amber	4	
			Search	Search Range from Station	Red	300	
				Search Response Time	Red	8	
				Search Weather Conditions	Green	4	

3.8 C8 – Planning Assumptions

NAFv3: NONE

The C8 Viewpoint is concerned with identification and description of assumptions that have been made for the implementation of capabilities.

Views implementing this Viewpoint:

- Shall contain capabilities relevant for the architecture.
- Shall include constraints for capability implementation.
- May include goals.
- May include assumed benefits.

CONCERNS ADDRESSED	USAGE
• Capability Planning. • Planning Assumptions.	• Implementation Planning.

REPRESENTATION

- Tabular.
- Benefits diagram

EXAMPLE

Example below shows a tabular representation of capabilities with associated constraints

Figure 3-8: Example C8 View

Capability	Constraint
Monitoring	Availability: 24x7x365
Recovery	Maximum Range: 1000 nm Concurrent Missions: 2
Search	Maximum Range: 1000 nm Concurrent Missions: 4

3.9 Cr– Capability Roadmap

NAFv3: NCV-3

The Cr Viewpoint is concerned with the representation of the actual or estimated availability of capabilities over a period of time (derived from capability delivery milestones in acquisition projects).
Views implementing this Viewpoint:

- Shall identify capabilities related to the roadmap.
- Shall identify associated capability increments.
- May identify programmes or projects associated with the capability increments.
- May associate capability increments with specific periods of time.

CONCERNS ADDRESSED	USAGE
• Capability Planning. • Acquisition Management.	• Capability phasing. • Capability integration planning. • Capability gap/surplus analysis. • High-level dashboard for acquisition management.

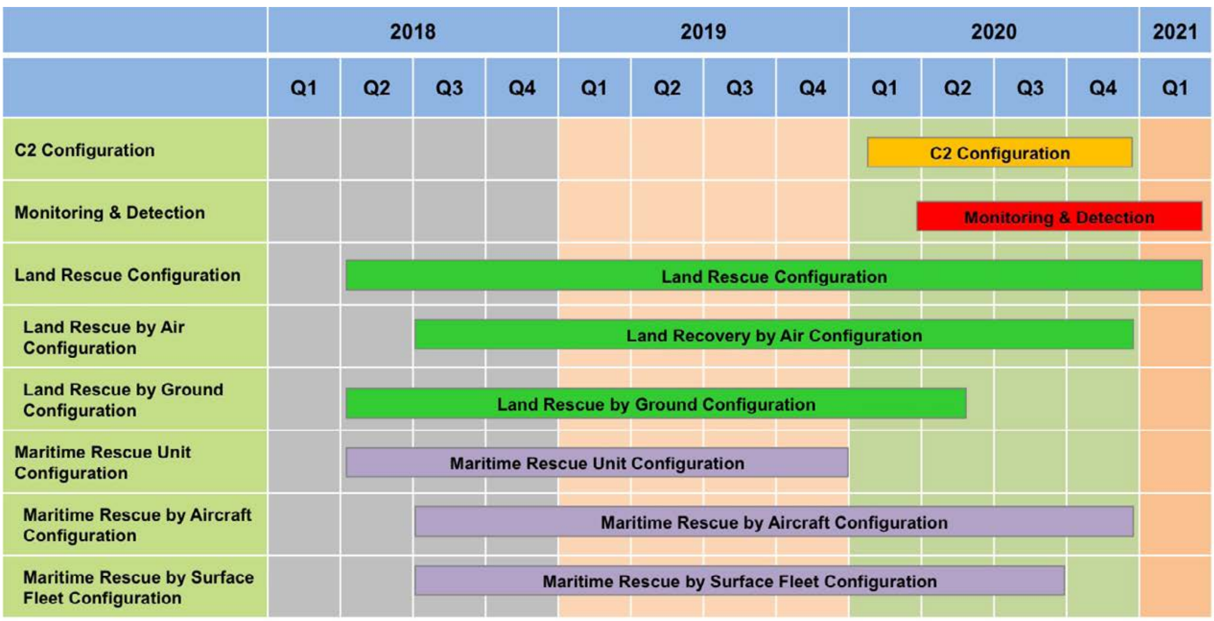
REPRESENTATION

- A time based chart in the style of a Gantt chart.

EXAMPLE

Below example (Figure 3-9) shows a timing chart with capabilities on the vertical axis and time on the horizontal axis. Active capability configurations are shown as bars against the capabilities they provide, with the start and end of the bars corresponding to the capability configuration coming into and going out of service. Where nothing meets a particular capability at a particular time, whitespace is shown so as to highlight capability gaps.

Figure 3-9: Example Cr View



4 SERVICE SPECIFICATION VIEWPOINTS

	Taxonomy	Structure	Connectivity	Processes	States	Sequences	Information	Constraints	Roadmap
	C1-S1 (NSOV-3)								
Service Specifications	S1 Service Taxonomy NAV-2, NSOV-1		S3 Service Interfaces NSOV-2	S4 Service Functions NSOV-3	S5 Service States NSOV-4b	S6 Service Interactions NSOV-4c	S7 Service I/F Parameters NSOV-2	S8 Service Policy NSOV-4a	Sr Service Roadmap

The Viewpoints in the Service Specifications row of the NAF grid support the description of services independently of how they are implemented or used. A service is understood in its broadest sense as a unit of work through which a provider provides a useful result to a consumer.

The purpose of these Viewpoints is to establish a library of standard services that support building architectures based on the concept of a service-oriented architecture. The Service Specifications Viewpoints describe services needed to directly support the operational domain.

The Service Specifications Viewpoints strictly focus on identifying and describing services, and does not specify their physical implementation (see Physical Resource Specifications layer). The Service Specifications layer also supports the description of service taxonomies, interfaces, policy and behaviour.

4.1 S1 – SERVICE TAXONOMY

NAFv3: NSOV-1/NAV-2

The S1 Viewpoint is concerned with the identification of service specifications, and their organization into specialization hierarchies (taxonomies).

Views implementing this Viewpoint:

- Shall include all service specifications relevant for the architecture.
- May organize all service specifications into a specialization hierarchy.
- May include measures for the service specifications.
- May include attributes for the service specifications.

A service taxonomy, in whole or parts, may be referenced by, or used in describing, multiple architectures (e.g. a S1 View at enterprise-level will be referenced by S1 Views at the capability-level).

CONCERNS ADDRESSED	USAGE
• Cataloguing Service Specifications. • Defining attributes used to measure Service Levels. • Specialization of Service Specifications.	• Service-oriented architecture governance. • Identification of services. • Service planning. • Service audit. • Service gap analysis. • Providing reference services for architectures. • Tailoring generic services for specific applications.

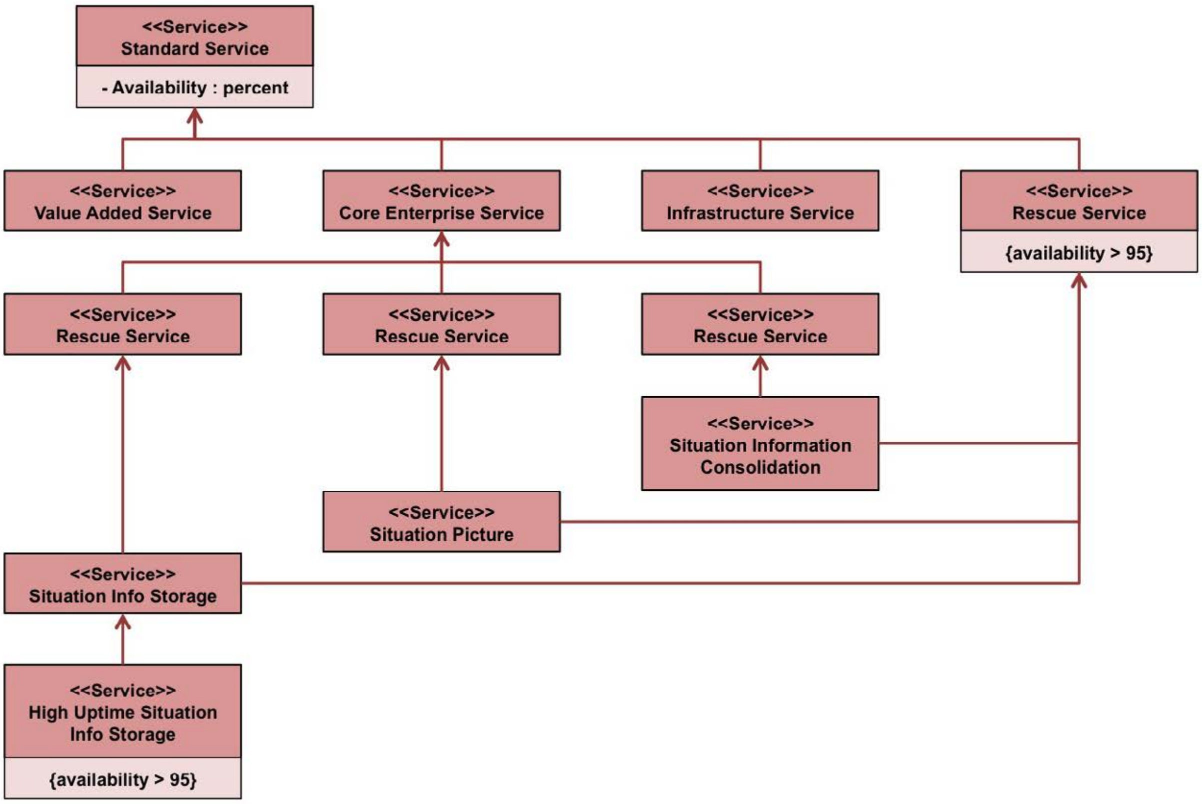
REPRESENTATION

- Tabulation.
- Hierarchical (connected shapes).
- UML class diagram.

EXAMPLE

The example in Figure 3-10 shows a taxonomy of Standard Services. There is also an availability attribute defined against the top service specification. All other service specifications inherit that attribute, and the Warfighting Service sets a constraint (service policy) that the availability shall be greater than 95%.

Figure 3-10: Example S1 View



4.2 S2 – Not Used

4.3 S3– Service Interfaces

NAFv3: NSOV-2

The S3 Viewpoint is concerned with the identification and definition of service interfaces.
Views implementing this Viewpoint:

- Shall identify service interfaces provided by a service.
- May identify service interfaces required by a service.
- May identify operations for service interfaces.
- May specify service operations.

CONCERNS ADDRESSED	USAGE
• Detailed Service Specifications. • Requirements for Service compatibility. • Service implementation guidance.	• Service-oriented architecture governance. • Detailed Service specification. • Service interoperability.

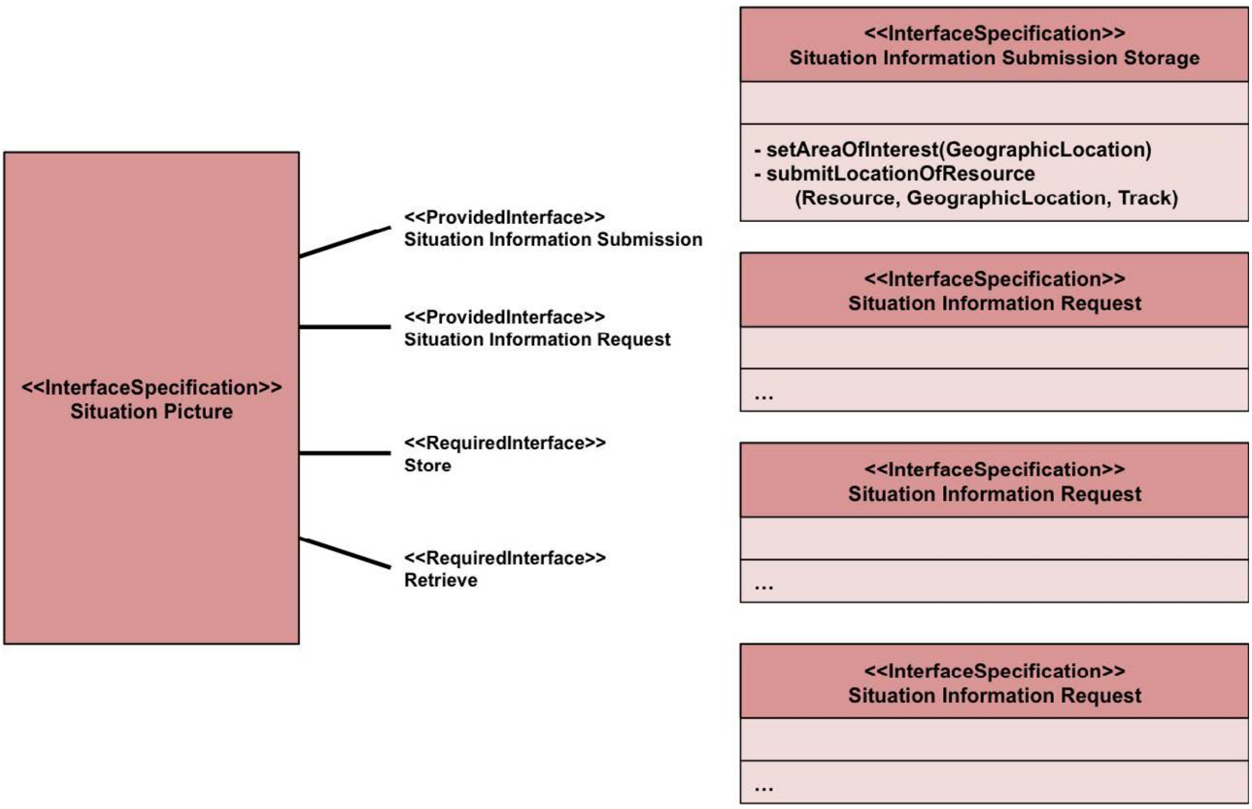
REPRESENTATION

- Tabular.
- UML.

EXAMPLE

Example below shows a tabular representation of capabilities with associated constraints.

Figure 3-11: Example S3 View

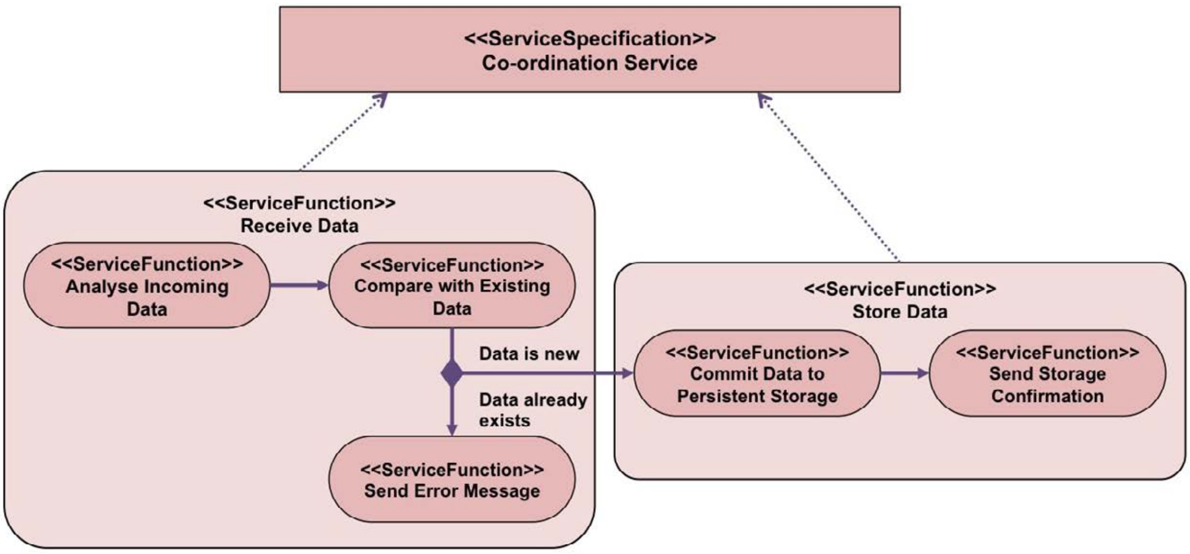


4.4 S4 – Service Functions	**NAFv3: NSOV-3**
The S4 Viewpoint is concerned with the definition of the behaviour of a service in terms of the functions it is expected to perform. Views implementing this Viewpoint: • Shall identify all functions a service is performing. • May specify composition of service functions.	
CONCERNS ADDRESSED	**USAGE**
• Detailed Service Specifications. • Outline requirements for Service behaviour. • Service implementation guidance.	• Service specification & planning. • Governance.
REPRESENTATION	
• Tabular. • UML.	

EXAMPLE

Example below shows a service specification with two functions, Receive Data and Store Data. Both these functions do have respective sub functions.

Figure 3-12: Example S4 View



4.5 S5– Service States

NAFv3: NSOV-4B

The S5 Viewpoint is concerned with the identification and definition of the possible states a service may have, and the possible transitions between those states.
Views implementing this Viewpoint:

- Shall identify and define all allowable states of a service.
- May describe possible state transitions.
- May describe service constraints.

CONCERNS ADDRESSED	USAGE
• Detailed Service Specifications. • Outline requirements for Service behaviour. • Service implementation guidance.	• Service behaviour specification.

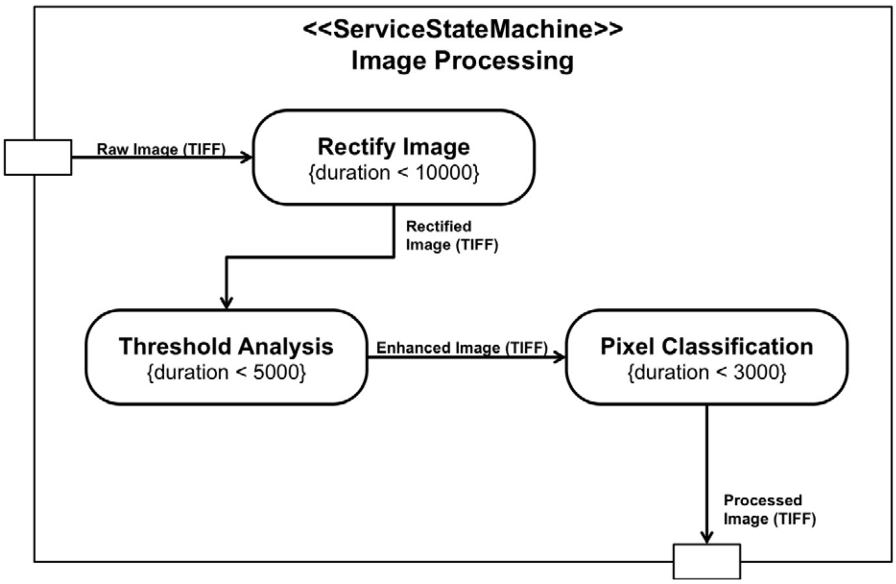
REPRESENTATION

- UML.
- Other state transition models.

EXAMPLE

Example below shows a state transition models including performance constraints (e.g. the maximum duration a service may be in a particular state).

Figure 3-13: Example S5 View



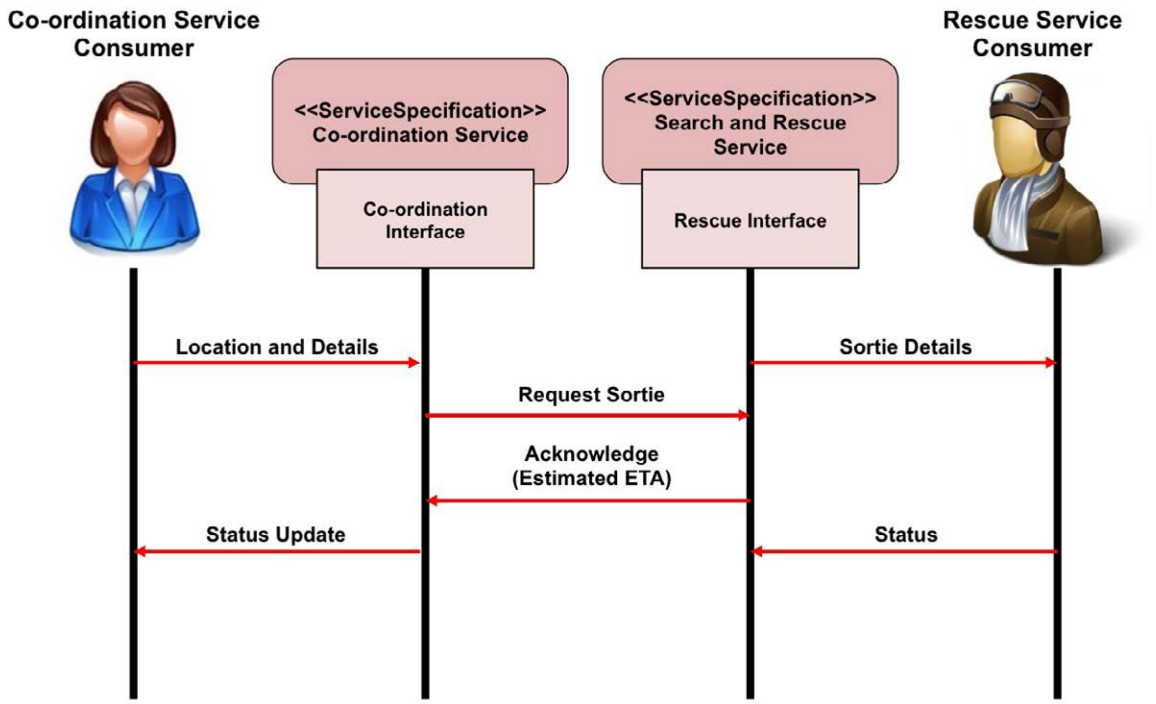
4.6 S6– Service Interactions	NAFv3: NSOV-4C
The S6 Viewpoint is concerned with describing interactions of a service with service consumers, and the sequence and dependencies of those interactions. Views implementing this Viewpoint: • Shall identify service is scope. • Shall identify service consumers. • Shall identify interactions of service consumers with the service. • May show service operations, and sequence of service operations. • May show service functions, and sequence of service functions.	

CONCERNS ADDRESSED	USAGE
• Detailed Service Specifications. • Outline requirements for Service behaviour. • Service implementation guidance.	• Service specification.

REPRESENTATION
• Sequence Diagram

EXAMPLE
Example below shows a service, two service consumers and their interactions.

Figure 3-14: Example S6 View



4.7 S7– Service Interface Parameters

NAFv3: NSOV-4C

The S7 Viewpoint is concerned with identification and specification of all the parameters used in service operations.

Views implementing this Viewpoint:

- Shall identify parameters for service operations relevant for the architecture.
- May specify the data types of each parameter.
- May show the assignment of service operations to service interfaces.

CONCERNS ADDRESSED	USAGE
• Detailed Service design. • Service compatibility analysis.	• Service-oriented architecture governance. • Detailed service specification. • Service interoperability.

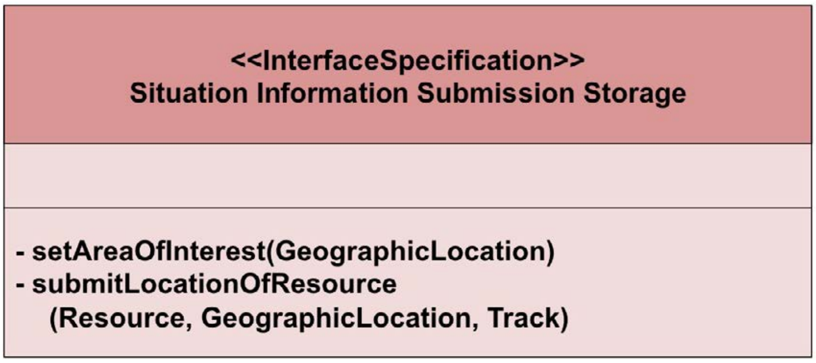
REPRESENTATION

- Tabular.
- UML.

EXAMPLE

Example below shows an Interface Specification with two operations, one operation (setAreaOfInterest) with one parameter, and a second operation (submitLocationOfResource) with three parameters.

Figure 3-15: Example S7 View



4.8 S8– Service Policy

NAFv3: NSOV-4C

The S8 Viewpoint is concerned with the identification and description of constraints that apply to service implementations.

Views implementing this Viewpoint:

- Shall define constraints that shall apply for implementations of each service specifications relevant for the architecture.
- May include measures for the service specifications.
- May include attributes for the service specifications.

CONCERNS ADDRESSED	USAGE
• Service Specifications. • Contracting for Services. • User / System Requirements.	• Service design. • Service governance.

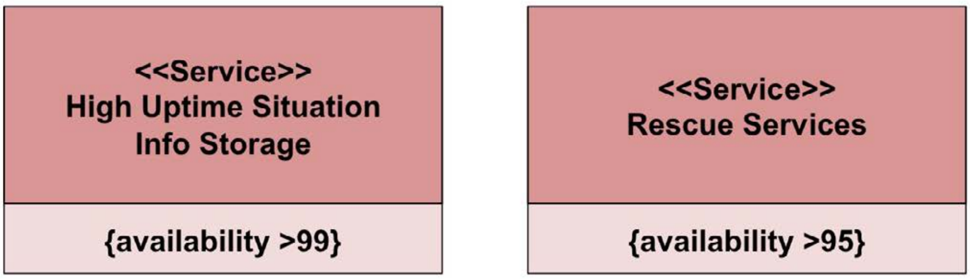
REPRESENTATION

- Tabular.
- UML.

EXAMPLE

Example below shows availability constraints for implementations of services.

Figure 3-16: Example S8 View



4.9 Sr– Service Roadmap

NAFv3: NONE

The Cr Viewpoint is concerned with the identification and description of life cycle information of service specifications.

Views implementing this Viewpoint:

- Shall identify service specifications related to the roadmap
- Shall define start and end date of service specification support.
- May identify programmes or projects associated with the service specification delivery/withdrawal.
- May identify service levels.
- May identify service attributes.
- May associate measures to service attributes.

CONCERNS ADDRESSED

- Service Life Cycle Planning.
- Acquisition Management.

USAGE

- Service phasing.
- Service gap/surplus analysis.
- High-level dashboard for acquisition management.

REPRESENTATION

- A time based chart in the style of a Gantt chart.
- Tabular.

EXAMPLE

Example below shows a tabular representation for In Service and Out of Service dates for a set of service specifications.

Figure 3-17: Example Sr View

Service Specification	In Service	Out of Service
Service A 1.0	May-18	Apr-19
Service A 2.0	Apr-19	Apr-21
Service B 2.0	Jan-18	Jul-19
Service B 3.0	Jun-19	Dec-21

4.10 C1-S1 – Capability to Service Mapping	NAFv3: NSOV-3

The C1-S1 Viewpoint is concerned with identification and description of services that enable capabilities. Views implementing this Viewpoint:

- Shall contain service specifications relevant for the architecture.
- Shall contain capabilities relevant for the architecture.
- Shall associate services to capabilities they enable.

CONCERNS ADDRESSED	USAGE
• Wrapping capabilities as Service Specifications. • Defining the capability required from a given Service Specification.	• Service Specification & Planning. • Governance.

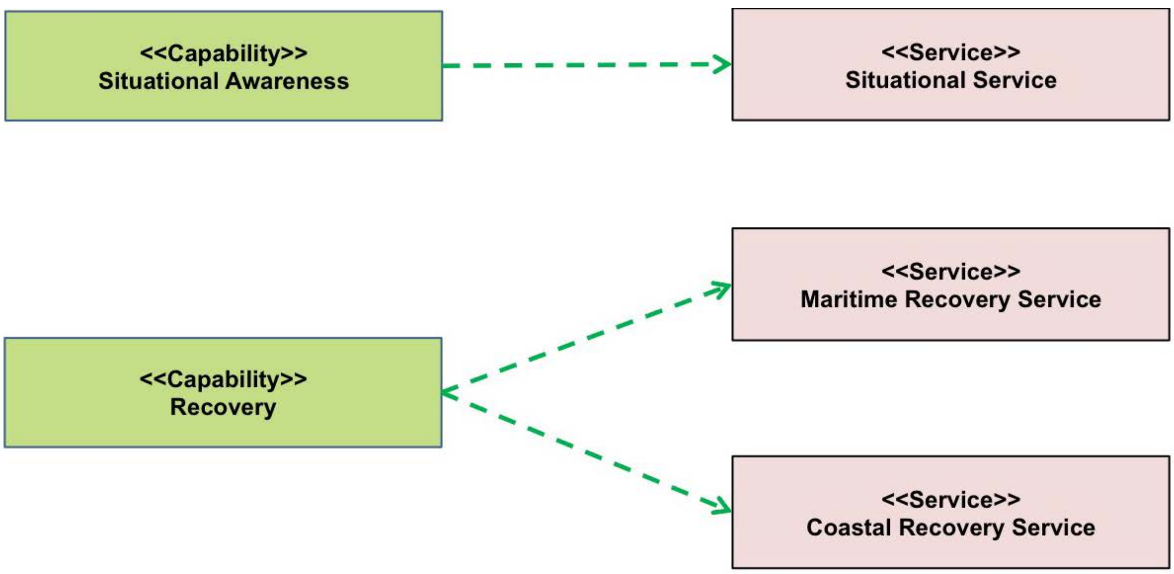
REPRESENTATION

- Matrix (with capabilities on one axis, and services on the other one).
- UML.

EXAMPLE

A C1-S1 can be presented as a diagram showing tracing relationships from capabilities to services.

Figure 3-18: Example C1-S1 View



5 LOGICAL SPECIFICATION VIEWPOINTS

	Taxonomy	Structure		Connectivity	Processes	States	Sequences	Information	Constraints	Roadmap
Logical Specifications	L1 Node Types NAV-2	L2 Logical Scenario NOV-2	L2-L3 (NOV-1)	L3 Node Interactions NOV-2, NOV-3	L4 Logical Activities NOV-5	L5 Logical States NOV-6b	L6 Logical Sequence NOV-6c	L7 Logical Data Model NSV-11a	L8 Logical Constraints NOV-6a	Lr Lines of Development NPV-2

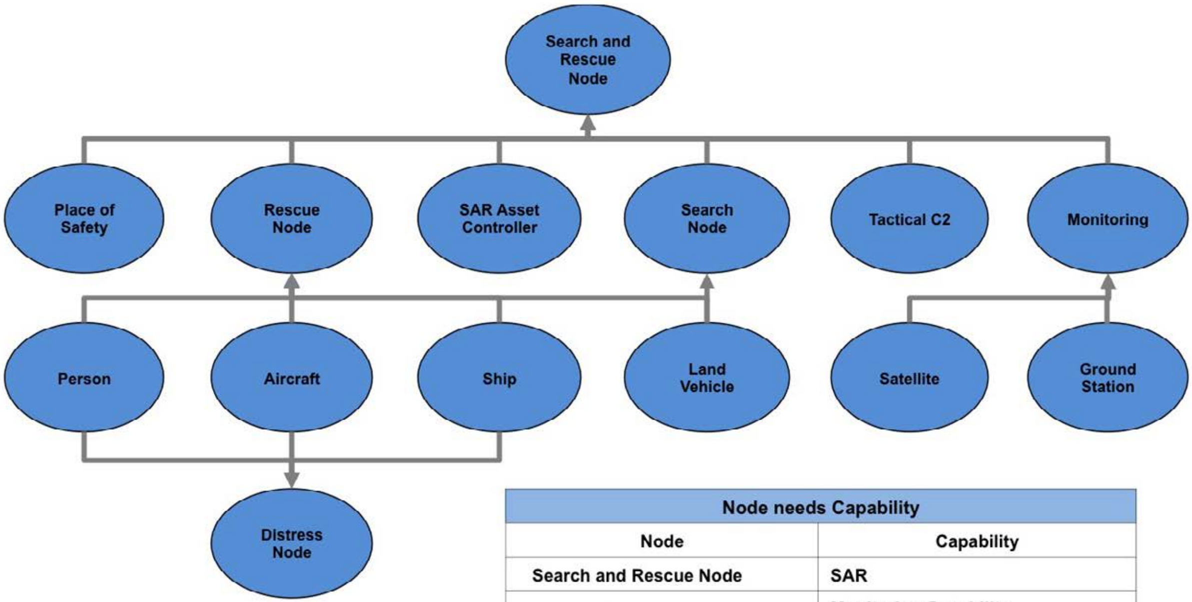
The Viewpoints in the Logical Specifications row of the NAF grid support the solution-independent description of the logical nodes (elements of capability), activities, and resource/information exchanges required to accomplish missions. Those missions include both war-fighting missions and business processes. The Logical Specifications Viewpoints specify graphical and textual Views that identify the logical nodes, their behaviour and interactions. Viewpoints in the Logical Specifications row address the specification of logical information (and resource) exchanges, the frequency of exchange and which activities produce and consume the exchanges (L3 Viewpoint). In addition, they address the specification of required service levels (L8 Viewpoint) and orchestration of services to support the mission (L6 Viewpoint).

5.1 L1– Node Types	NAFv3: NAV-2
The L1 Viewpoint is concerned with the identification of nodes, and their organization into specialization hierarchies (taxonomies). In the NAF, nodes are logical entities (i.e. defined independent of their implementation) that are able to perform behaviour. Views implementing this Viewpoint: • Shall identify all nodes relevant for the architecture. • May show a specialization hierarchy for nodes. • May trace nodes to capabilities they need. • May trace nodes to roles they are performing in activities. • May include Measures of Performance (MoP). A node taxonomy, in whole or parts, may be referenced by, or used in describing, multiple architectures (e.g. a L1 View at enterprise-level will be referenced by L1 Views at the capability-level).	
CONCERNS ADDRESSED	**USAGE**
• User Requirements. • Operational Planning. • High-Level Systems Requirements.	• Initial set up of a Logical Architecture. • Defining MoP for requirements specification purposes. • Defining the types of environment in which Nodes may operate.
REPRESENTATION	
• Topological (connected shapes). • Tabular.	

EXAMPLE

Example below shows nodes arranged in a specialization hierarchy in a hierarchy diagram. It also shows a table to trace capabilities to nodes.

Figure 3-19: Example L1 View



Node needs Capability	
Node	Capability
Search and Rescue Node	SAR
Distress Node	Monitoring Capability Distress Monitoring Signal
Place of Safety	Assistance
Rescue Node	Recovery Capability Recovery Coastal Capability Recovery Land Capability Recovery Water Capability Recovery Inland Water Capability Recovery Maritime Capability Recovery Coastal Capability

5.2 L2 – Logical Scenario

NAFv3: NOV-2

The L2 Viewpoint is concerned with identifying key or aggregated interactions between nodes.
Views implementing this Viewpoint:

- Shall include nodes relevant for the architecture.
- Shall define logical flows (e.g. logical flow of information) independent of their implementation.
- Shall only include key individual and/or aggregated logical flows between nodes.
- May include a mapping of nodes to locations.

CONCERNS ADDRESSED	USAGE
• User Requirements. • Operational Planning. • Scenario Specification.	• Definition of operational concepts. • Elaboration of capability requirements. • Definition of collaboration needs. • Associating capability with a location. • Problem space definition. • Operational planning. • Supply chain analysis. The L2 Viewpoint can be enhanced with additional features for modelling security: • Security domain specification. • Logical entity trust models. • Threat specification (e.g. threat vectors) and counter-capability specifications.

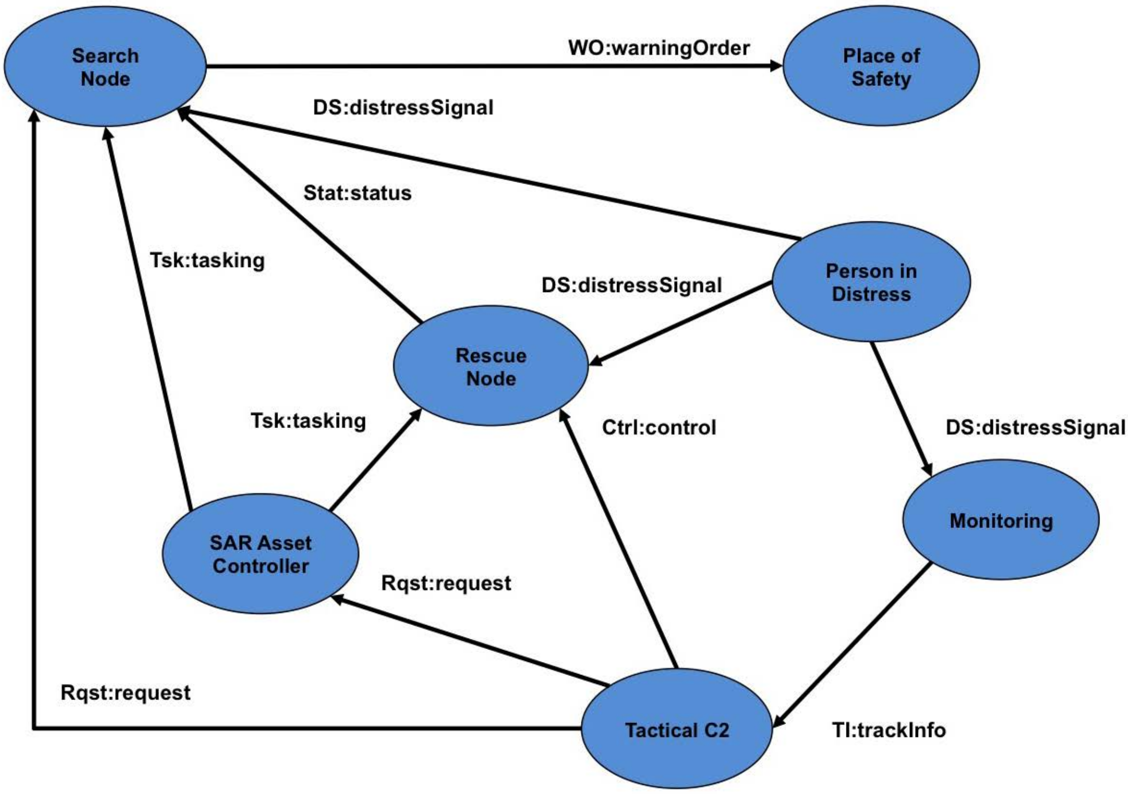
REPRESENTATION

- Topological (connected shapes).
- UML composite structure diagram.

EXAMPLE

Figure 3-20 shows interactions between nodes, in this case flows of information between the nodes.

Figure 3-20: Example L2 View



5.3 L3 – Node Interactions

NAFv3: NOV-2, 3

The L2 Viewpoint is concerned with identifying all relevant interactions between nodes.
Views implementing this Viewpoint:

- Shall include nodes relevant for the architecture.
- Shall include all logical flows (e.g. logical flow of information) between nodes relevant to the architecture.
- Shall define logical flows independent of their implementation.
- May associate the logical flows to logical activities.
- May define properties of the logical flows.
- May define measure of the logical flows.

CONCERNS ADDRESSED

- Interoperability Requirements.

USAGE

- Definition of interoperability requirements.

REPRESENTATION

- Tabulation.
- Information flow diagram.

EXAMPLE

Example below shows a table detailing information flows with source and target node of the flow, and additional properties (e.g. media type) or measures (e.g. availability).

Figure 3-21: Example L3 View

	Purpose	Source	Target	Transaction Type	Media Type	Interoperability Level Required	Availability	Associated Collaborations	Information Element is conveyed by Information Exchange
Ctrl:control	control rescue	Tactical C2	Rescue Node		Voice	Level 3	95	Iex:Controls	Assignment Search Object
DS:distressSignal	signal rescue node	Person in Distress	Rescue Node		Voice			Iex:DistressSignal	Last Known Position
DS:distressSignal	signal monitoring station	Person in Distress	Monitoring					Iex:DistressSignal	Last Known Position
DS:distressSignal	signal search node	Person in Distress	Search Node					Iex:DistressSignal	Last Known Position
Rqst:request	request search	Tactical C2	Search Node					Iex:Request	SAR Operation
Rqst:request	asset required	Tactical C2	SAR Asset Controller					Iex:Request	SAR Operation
Stat:status	status update	Rescue Node	Search Node					Iex:Status	Search Status
TI:trackingInfo	tracking information	Monitoring	Tactical C2					Iex:Tracking	Last Known Position
Tsk:tasking	task search	SAR Asset Controller	Search Node					Iex:Tasking	Assignment
Tsk:tasking	task rescue	SAR Asset Controller	Rescue Node					Iex:Tasking	Assignment
WO:warningOrder	warning	Search Node	Place of Safety					Iex:WarningOrder	Search Object

5.4 L4 – Logical Activities

NAFv3: NOV-5

The L4 Viewpoint is concerned with the identification of logical (i.e. implementation independent) activities, grouping and composition of these activities, and logical flows between the activities.
Views implementing this Viewpoint:

- Shall identify logical activities relevant for the architecture.
- May identify groupings of activities.
- May identify composition of activities.
- May associate logical activities to nodes.
- May identify logical flows between activities.

CONCERNS ADDRESSED	USAGE
• Process Modelling. • Operational Planning. • Concept of Operations. • Service Orchestration.	• Requirements capture. • Description of business processes and workflows. • Operational planning. • Logistics support analysis. • Information flow analysis. • Support task analysis to determine training needs.

REPRESENTATION

- Hierarchy chart.
- IDEF0 Diagram.
- UML activity diagram.
- UML activity diagram (with swim-lanes).

EXAMPLE

Example below shows a tabular representation for In Service and Out of Service dates for a set of service specifications.

Figure 3-22: Example L4 View

5.5 L5 – Logical States

NAFv3: NOV-16B

The L5 Viewpoint is concerned with the identification and definition of the possible states a node may have, and the possible transitions between those states.
Views implementing this Viewpoint:

- Shall identify and define all states of a node relevant for the architecture.
- May describe possible state transitions.

CONCERNS ADDRESSED	USAGE
• Scenario Specification. • User Requirements Specification.	• Analysis of business events. • Behavioural analysis. • Identification of constraints.

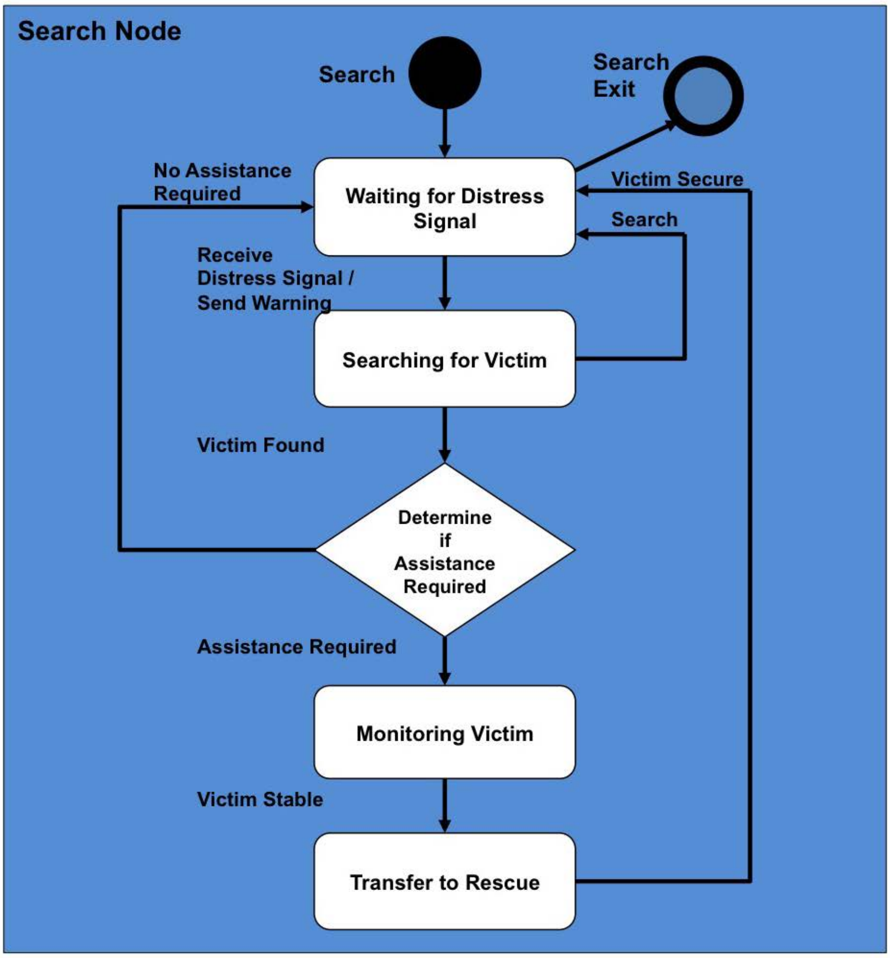
REPRESENTATION

- Topological (Connected Shapes).
- UML state diagram.

EXAMPLE

Example below show a simple state transition diagram with four states and transitions between them.

Figure 3-23: Example L5 View



5.6 L6 – Logical Sequence

NAFv3: NOV-6C

The L6 Viewpoint is concerned with identifying and describing the chronological sequence of activities and/or logical flows in a scenario.
Views implementing this Viewpoint:

- Shall identify the activities and/or logical flows relevant for a scenario.
- Shall identify the chronological sequence of activities and/or logical flows.
- May identify source and target nodes of logical flows
- May identify start and end events of a sequence.

CONCERNS ADDRESSED	USAGE
• Operational Planning. • User Requirements Specification. • Service Orchestration.	• Analysis of operational events. • Sequences of interactions between nodes. • Behavioural analysis. • Identification of non-functional user requirements. • Operational test scenarios.

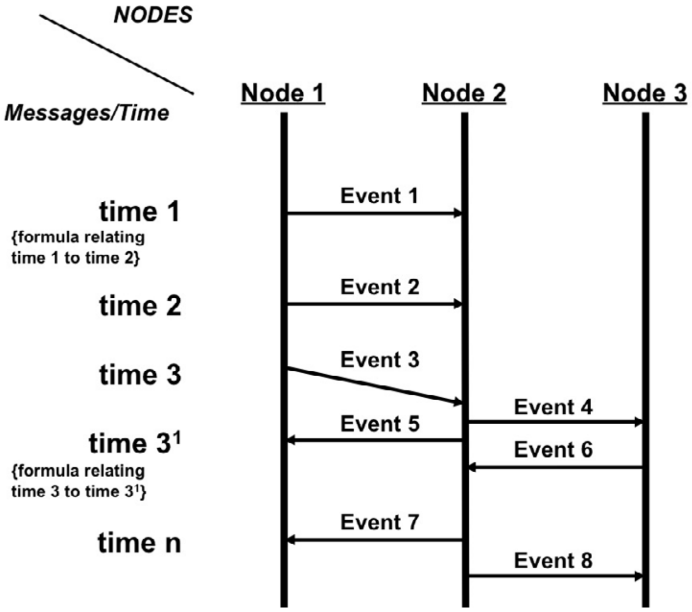
REPRESENTATION

- UML sequence diagram.
- Event-trace diagram.
- Timing diagram.

EXAMPLE

Example below shows the logical sequence of interaction between three nodes, and associated events.

Figure 3-24: Example L6 View



5.7 L7 – Logical Data Model

NAFv3: NOV-7

The L7 Viewpoint is concerned with identifying information elements, and describing their relationships.
Views implementing this Viewpoint:
- Shall identify information elements relevant for the architecture.
- May identify relationships between information elements.
- May identify attributes of information elements.
- May associate attributes with data entities.

CONCERNS ADDRESSED	USAGE
• Information Requirements. • Message Requirements. • Logical Data Models.	• Information architecture. • Information product hierarchy.

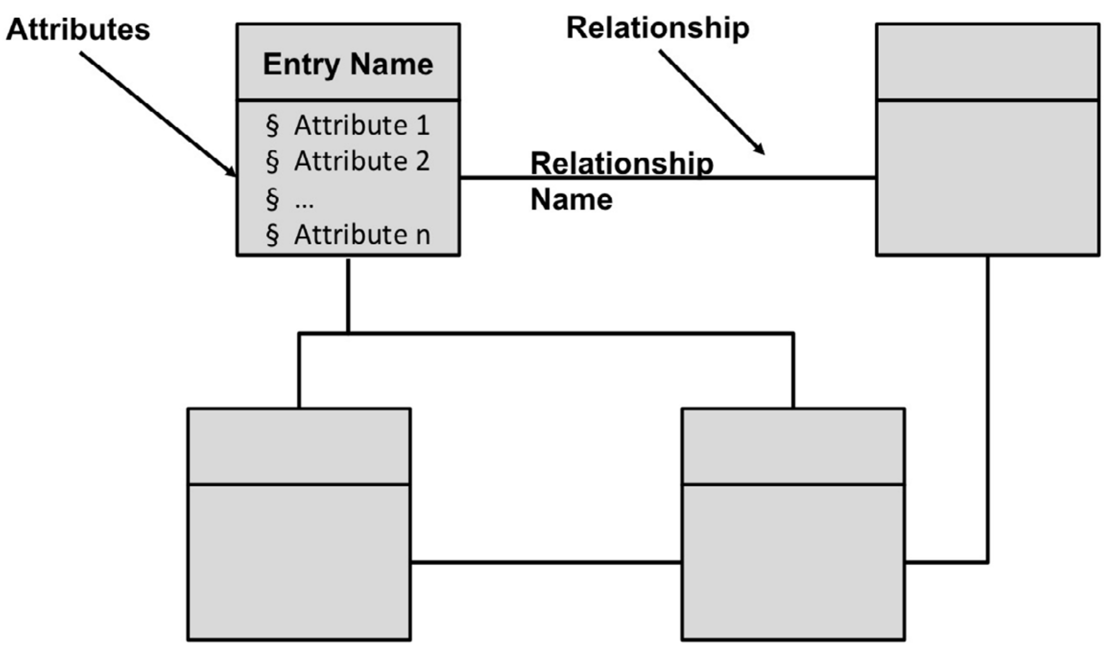
REPRESENTATION

- Entity-Relationship diagram (e.g. IDEF1X).
- UML class diagram.

EXAMPLE

Example below shows as generic class diagram with classes representing the information elements, and attributes of, as well as relationships between, these classes.

Figure 3-25: Example L7 View



5.8 L8 – Logical Constraints

NAFv3: NOV-6A

The L8 Viewpoint is concerned with identification and description of operational or business rules.
Views implementing this Viewpoint:

- Shall identify operational or business rules relevant for the architecture.
- Shall assign these rules to nodes, activities and/or logical flows.

CONCERNS ADDRESSED	USAGE
• User Requirements Specification (Non-Functional). • Operational Constraints.	• Definition of business rules. • Identification of operational constraints.

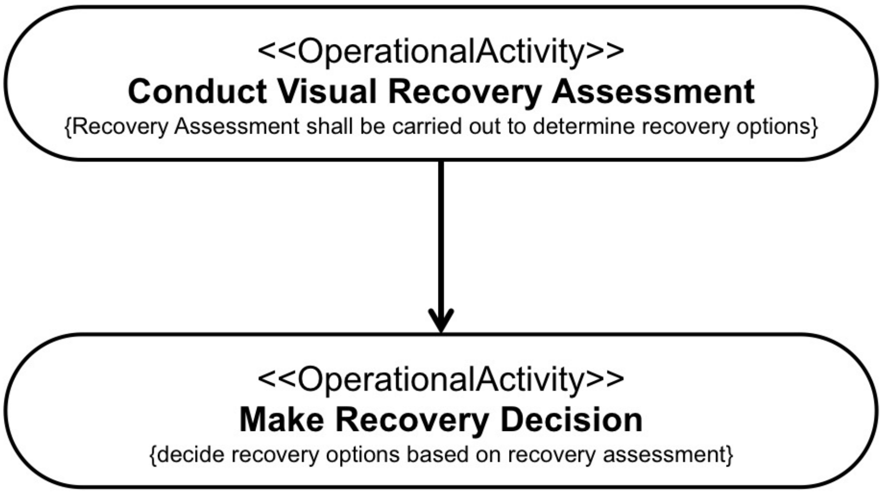
REPRESENTATION

- Structured Text.
- UML diagram with associated UML constraints.
- Business rules diagram.

EXAMPLE

Example below shows as generic class diagram with classes representing the information elements, and attributes of, as well as relationships between, these classes.

Figure 3-26: Example L8 View

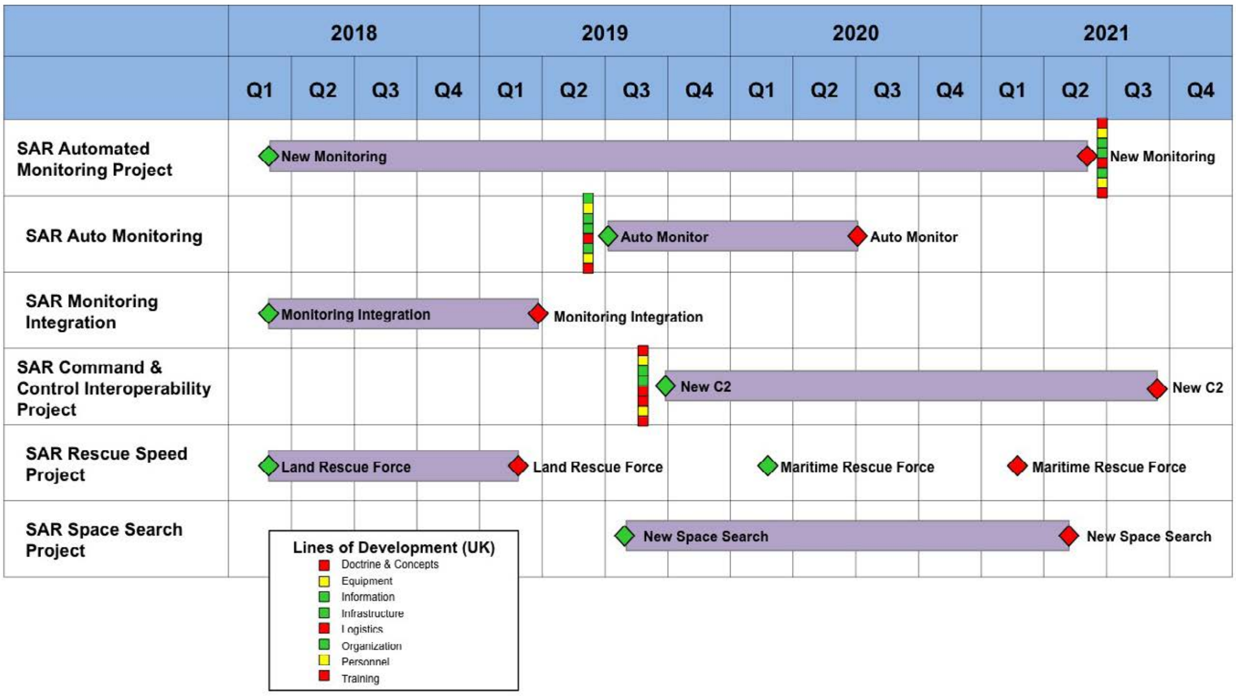


5.9 Lr – Lines of Development	NAFv3: NPV-2
The Lr Viewpoint is concerned with identifying and defining logical threads (lines of developments) for a set of projects and/or programmes. Views implementing this Viewpoint: • Shall identify project deliverables (e.g. capability increments or resource packages). • Shall associate project deliverables to project milestones. • May show states of deliverables at project milestones. • May associate project deliverables to enterprise phases. • May show project milestone dependencies.	
CONCERNS ADDRESSED	**USAGE**
• Acquisition Planning. • Portfolio / Programme Management. • Project Performance Reporting / Dash boarding.	• Project management and control (including delivery timescales). • Project dependencies and the identification of associated risk. • Portfolio management. • Through Life Management Planning.
REPRESENTATION	
• Timeline View. • Augmented Gantt Chart.	

EXAMPLE

Example below shows a roadmap for several capabilities. Status bars at specific events show the status of the capability against all lines of development.

Figure 3-27: Example Lr View



5.10 L2-L3 – Logical Concept Viewpoint

NAFv3: NOV-1

The L2-L3 Viewpoint is concerned with providing an executive level, scenario-based communication of the architecture purpose, scope and content.

A View implementing this Viewpoint:

- Shall show the main elements in scope of the Architecture Description.
- Shall show the main interactions of these elements.
- May show interactions of the main elements with elements outside the scope.
- May include any meta-model element.
- May include rich picture or graphics.

CONCERNS ADDRESSED	USAGE
• High-Level Communication of Architecture. • Senior Stakeholder Engagement.	• Puts an operational situation or scenario into context. • Provides a tool for discussion and presentation; e.g. aids industry engagement in acquisition. • Provides an overview of more detailed information in published architectures.

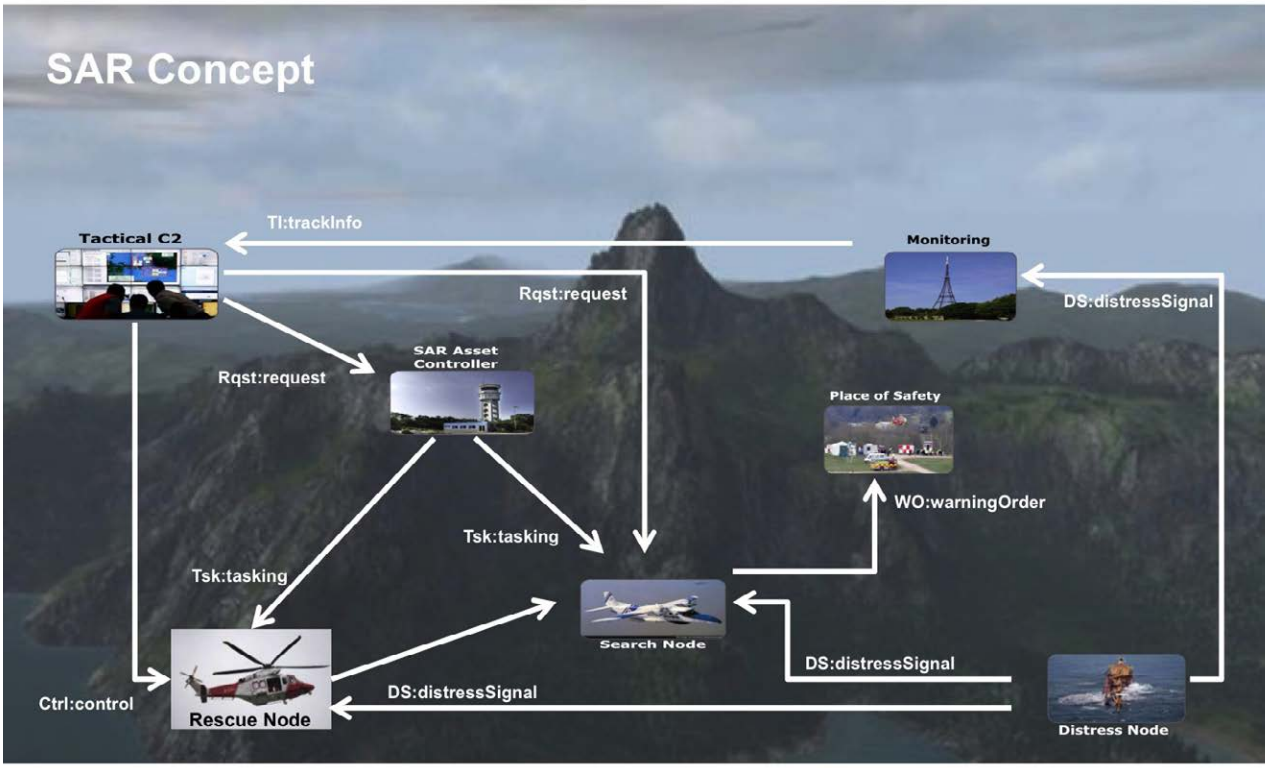
REPRESENTATION

- Graphic.
- Rich Picture.
- Concept diagram.
- Project context diagram.

EXAMPLE

Example below shows a concept diagram displaying the key elements (e.g. Tactical C2 and Monitoring) and interactions in a Search and Rescue Scenario.

Figure 3-28: Example L2-L3 View



6 PHYSICAL RESOURCE SPECIFICATION VIEWPOINTS

	Taxonomy	Structure	Connectivity	Processes	States	Sequences	Information	Constraints	Roadmap
				L4-P4 (NSV-5)					
Physical Resource Specifications	P1 Resource Types NAV-2, NSV-2a,7,9,12	P2 Resource Structure NOV-4,NSV-1	P3 Resource Connectivity NSV-2, NSV-6	P4 Resource Functions NSV-4	P5 Resource States NSV-10b	P6 Resource Sequence NSV-10c	P7 Physical Data Model NSV-11b	P8 Resource Constraints NSV-10a	Pr Configuration Management NSV-8

Viewpoints in the Physical Resource Specifications row of the NAF grid support the description of the structure, connectivity and behaviour of the various types of Resources. Resource Types include people, organizations, artefacts, software and configurations of any or all of them. In particular, these Viewpoints are used to specify how Types of Resources are configured and connected to deliver Capabilities and Services. The Physical Resource Specifications Viewpoints are used to support functions in both war-fighting and business. They can be used to link Resources back to the logical nodes specified in the Logical Specifications Viewpoints to provide requirements traceability. Resource Functions are also traced back to Activities in the Logical Specifications Viewpoints.

6.1 P1 - Resource Types

NAFv3: NAV-2/NSV-2A, 7, 9, 12

The P1 Viewpoint is concerned with specification of the types of resources and identifying required technologies and competences.

Views implementing this Viewpoint:

- Shall include all Resource Types relevant for the architecture together with a depiction of their performance characteristics.
- Shall describe the interface protocols and hardware specifications of each port on a system and include properties of Resource ports exposed by technical resources.
- Shall map the described Resource Types back to the Capabilities and/or Services they implement (without specifying these Services themselves).
- Shall provide a summary of the technologies and competences that impact on the Resources constituting the architecture.
- Shall specify Service Levels for the implemented Services and for other Services (effectively a composition of services) required for their implementation.
- May include descriptions of relevant emerging and current technologies, industry trends, predictions of the availability and readiness of specific hardware/software products, current and possible future skills.
- May organize the Resources into a specialization hierarchy.
- May give forecasts of relevant technologies and competences in short, mid and long-term timeframes and include an assessment of the potential impact of the forecast items on the enterprise.

CONCERNS ADDRESSED	USAGE
• Capability Delivery. • Service Implementation. • Interface Specification.	• Identifying Resource Taxonomies. • Interface specification. • Identification of applicable protocols. • Service implementation. • Tracing business processes to the resources that support them. • Forecasting technology readiness against time. • HR trends analysis. • Recruitment planning. • Planning technology insertion. • Input to options analysis. • Definition of performance characteristics. • Identification of non-functional requirements.

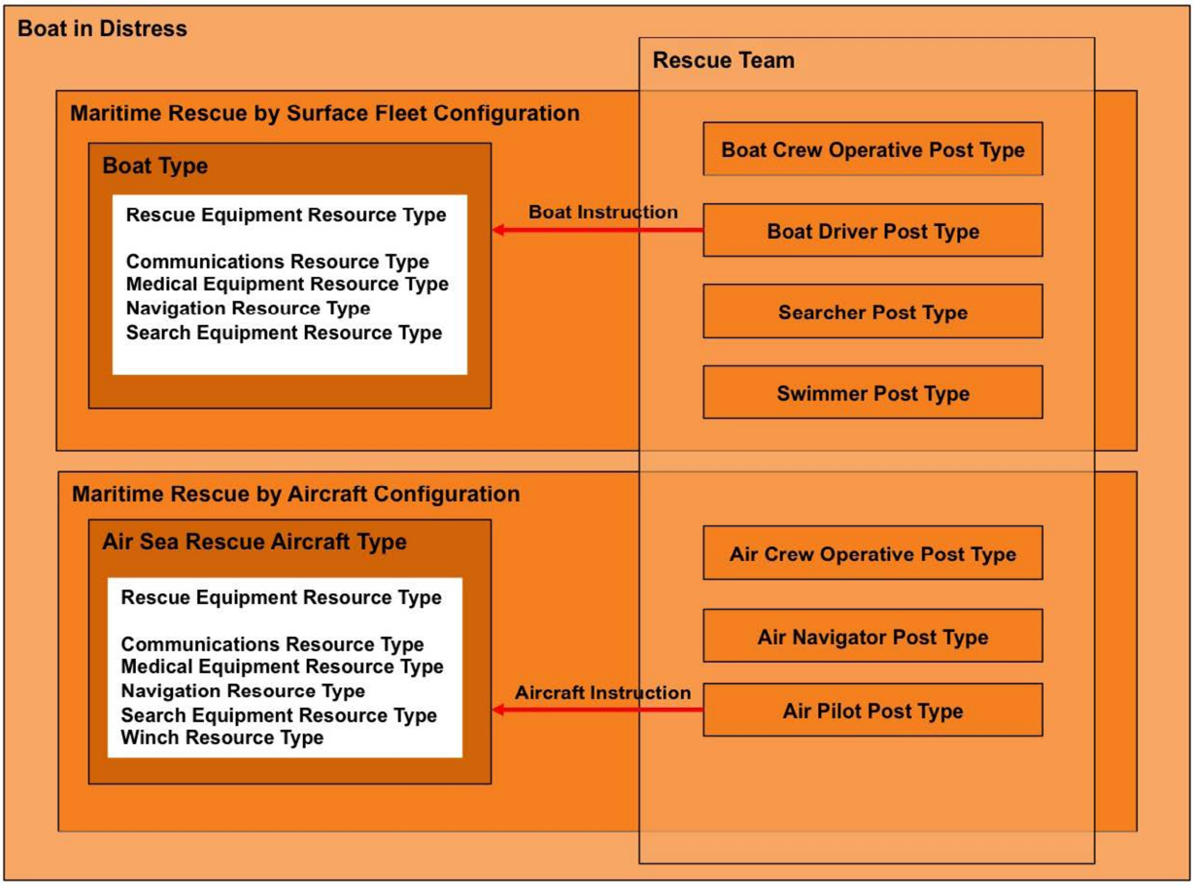
REPRESENTATION

- Tabulation.
- Mapping (matrix).
- Topological – connected shapes.
- UML Composite Structure Diagram.
- SysML block diagram.
- Timeline View.
- ‘Herringbone’ diagram.

EXAMPLE

Example below shows two possible configurations of a rescue unit, one based on a maritime vessel and a second one based on an aircraft.

Figure 3-29: Example P1 View



6.2 P2 – Resource Structure

NAFv3: NSV-1/NOV-4

The P2 Viewpoint is concerned with the composition and (high-level) interaction of resources.
Views implementing this Viewpoint:

- Shall link together the operational and physical Architecture Views by depicting how types of Resource are structured and interact to realize the logical architecture specified in L2, Logical Scenario.
- Shall describe the structure of resources, decomposed to any suitable level, by identifying the primary sub-systems, posts/roles and their interactions (e.g., data, materiel, human resources, energy).
- Shall gather systems meeting a specific capability as Capability Configurations.
- May represent the realisation of a requirement specified in a L2, maybe as several alternative Resource Views suites which could realize the operational requirement.
- May specify typical (or template) organizational structures, and also identify how human resources interact with each other and with systems.
- May identify the artefacts upon which resources are deployed and can show the nodes that the resources realize.

CONCERNS ADDRESSED	USAGE
• Physical Architecture. • Systems Engineering / Design. • Organizational Design. • Systems Integration. • System Requirements Specification.	• Definition of system concepts. • Definition of system options. • Human – System interactions. • Typical Organization structures. • Interface requirements capture. • Capability integration planning. • System integration management. • Operational planning (capability configuration definition).

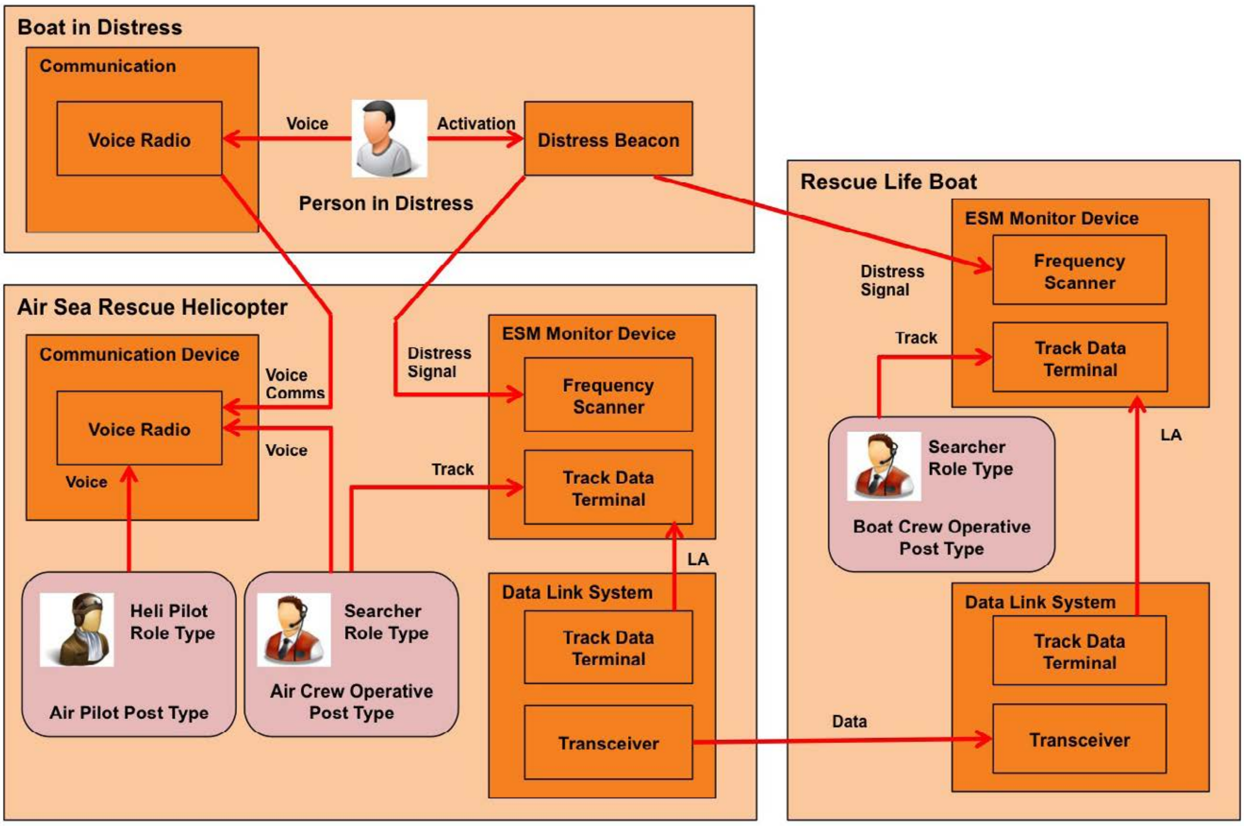
REPRESENTATION

- Topological (connected shapes).
- UML composite structure diagram (typical).
- UML instances (actual).
- SysML block diagram.

EXAMPLE

Example below shows the interaction between the individual resources in a search and rescue scenario.

Figure 3-30: Example P2 View



6.3 P3 – Resource Connectivity

NAFv3: NSV-2B, 2C, 6

The P3 Viewpoint is concerned with communication networks and pathways that link communications systems, details regarding their configuration and characteristics of the data exchanged between systems.
Views implementing this Viewpoint:

- Shall represent the physical implementation of the logical flows (L2, Logical Scenario, or L3, Node Interactions View) by specifying how systems are connected.
- Shall provide more technical detail than P2, including the protocols (specified in the P1 View) implemented by systems and used by the connections between those systems.
- Shall focus on the physical characteristics of each link by specifying attributes (e.g., geographic location, layout of network components such as routers, switches, amplifiers and repeaters).
- Shall include capacities (e.g. bandwidth, throughput), frequencies used, security encryption methods used and other descriptive information as attributes.
- Shall only feature physical architectures, software and artefacts (as systems) and no organizational resources.
- Shall show flows (as data elements relating to the P4, Resource Function Viewpoint) across system boundaries and no internal flows which so not correspond to system port connections.

CONCERNS ADDRESSED	USAGE
• Interface Specification. • Systems Engineering. • System Requirements.	• Interface specification. • Identification of applicable protocols. • Description of system communication paths. • Bandwidth and capacity analysis. • Detailed definition of data flows.

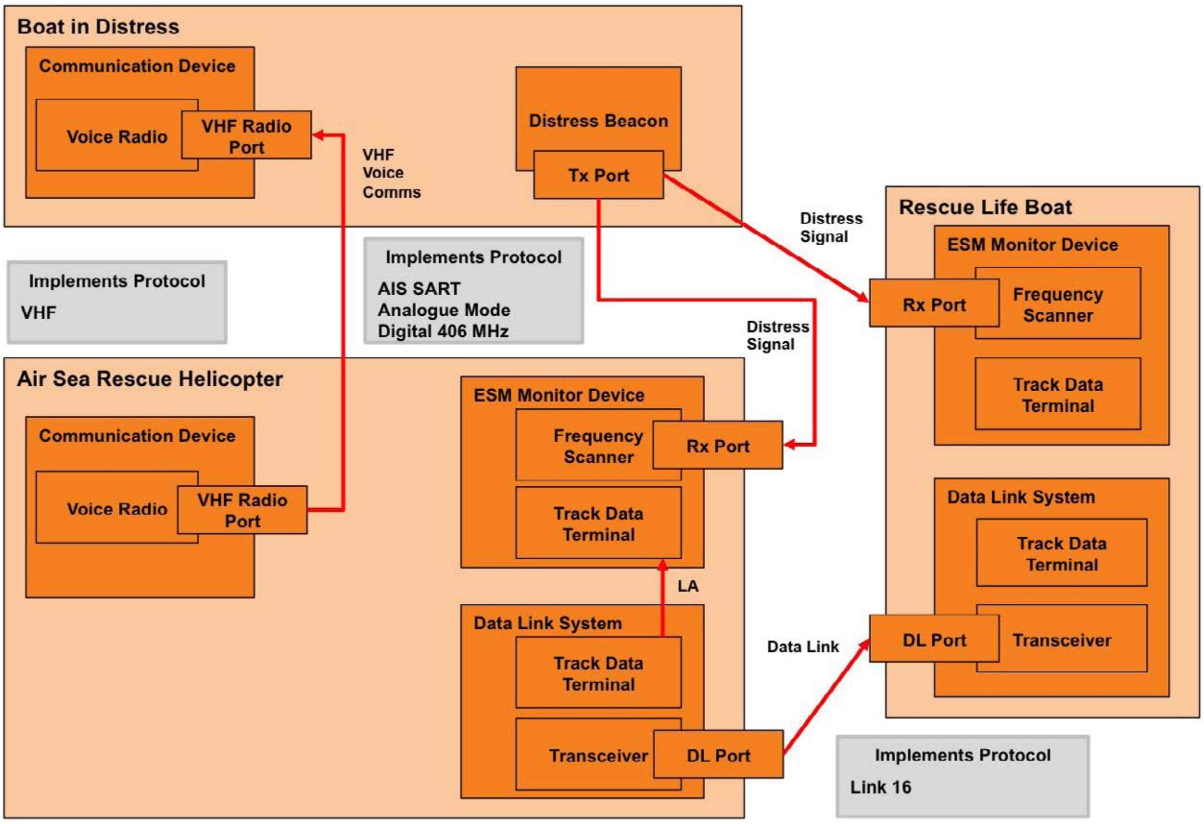
REPRESENTATION

- Topological (connected shapes).
- UML composite structure diagram.
- SysML structural diagram.
- Tabulation.

EXAMPLE

The example P3 View specifies the communications links between systems and may also list the protocol stacks used in connections. The architect may choose to create a diagram for each pair of connected systems in the architecture (see Figure 3-31 or to show all the connections on one diagram if this is possible.

Figure 3-31: Example P3 View



6.4 P4 – Resource Functions

NAFv3: NSV-4

The P4 Viewpoint is concerning the Resource Functions carried out by all types of Resource (human and non-human), including organizational resources.
Views implementing this Viewpoint:

- Shall specify the functionality of resources in the architecture as the functional counterpart to the structures specified in the P2, Resource Structure Views.
- Shall include detailed information regarding the allocation of functions to resources, and the flow of data between Resource Functions as the Physical Resource counterpart to the L4, Logical Activities Views.
- Shall describe implementation-specific realisations of the operational activities specified in the L4, Logical Activities Viewpoint.
- Shall include the complete functional connectivity (i.e. a resource's required inputs are all satisfied).

CONCERNS ADDRESSED	USAGE
• Capability-Based Acquisition. • Business Process Modelling. • Workflow Modelling. • Human-Machine Interaction Specifications.	• Description of task workflow. • Identification of functional system requirements. • Functional decomposition of systems. • Relate human and system functions.

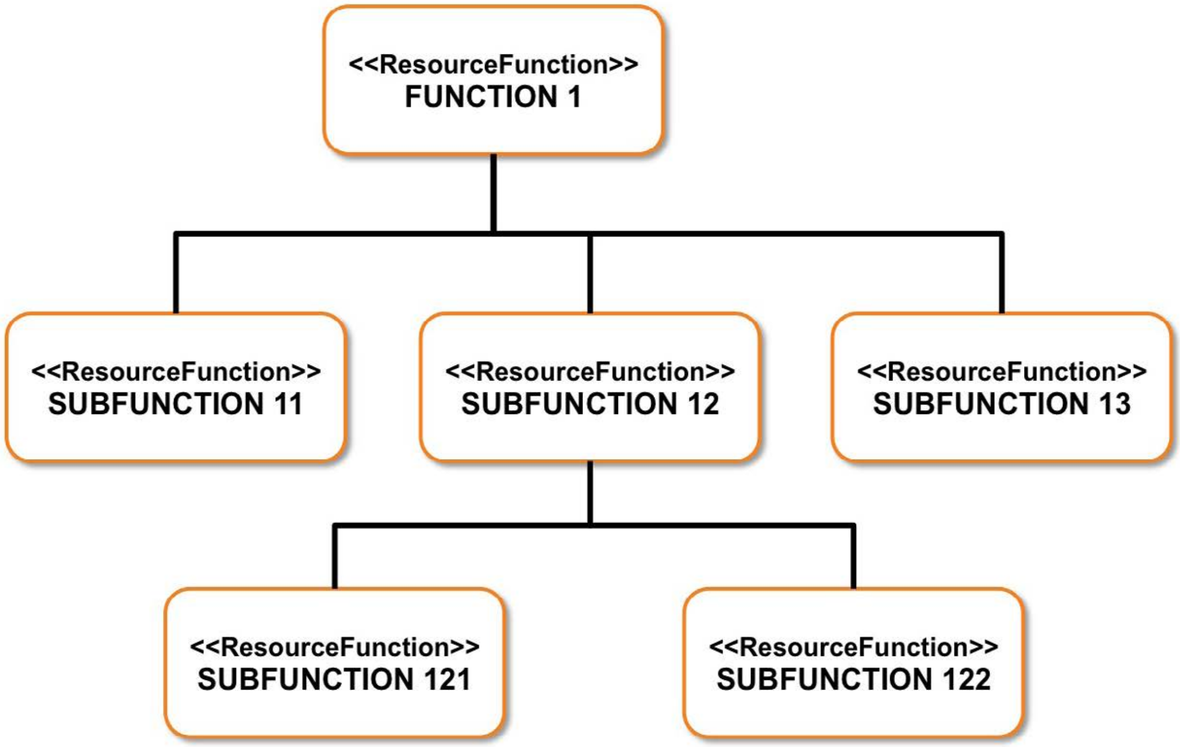
REPRESENTATION

- Topological (connected shapes).
- UML activity diagram.
- UML activity diagram (with swim lanes to represent resources).
- Functional Breakdown (decomposition).
- SysML activity diagram.

EXAMPLE

The example shows a decomposition of Resource Functions depicted in a tree structure and is typically used where tasks are concurrent but dependent, for example, on a production line.

Figure 3-32: Example P4 View



6.5 P5 – Resource States

NAFv3: NSV-10B

The P5 Viewpoint is concerned with Resource Types changing state in response to events and other stimuli. Views implementing this Viewpoint:

- Shall identify the states a Resource Type can be, the allowable changes between those states, and the triggers that cause the state changes.
- Shall relate events to Resource Type states and describe the transition from one state to another from a resource perspective, with a focus on how the Resource Type responds to stimuli (e.g. triggers and events).
- May describe different responses depending upon the rule set or conditions that apply, as well as the Resource Type's state at the time the stimuli is received.

CONCERNS ADDRESSED	USAGE
• Systems Engineering. • Safety Cases.	• Definition of states, events and state transitions (behavioural modelling). • Identification of constraints.

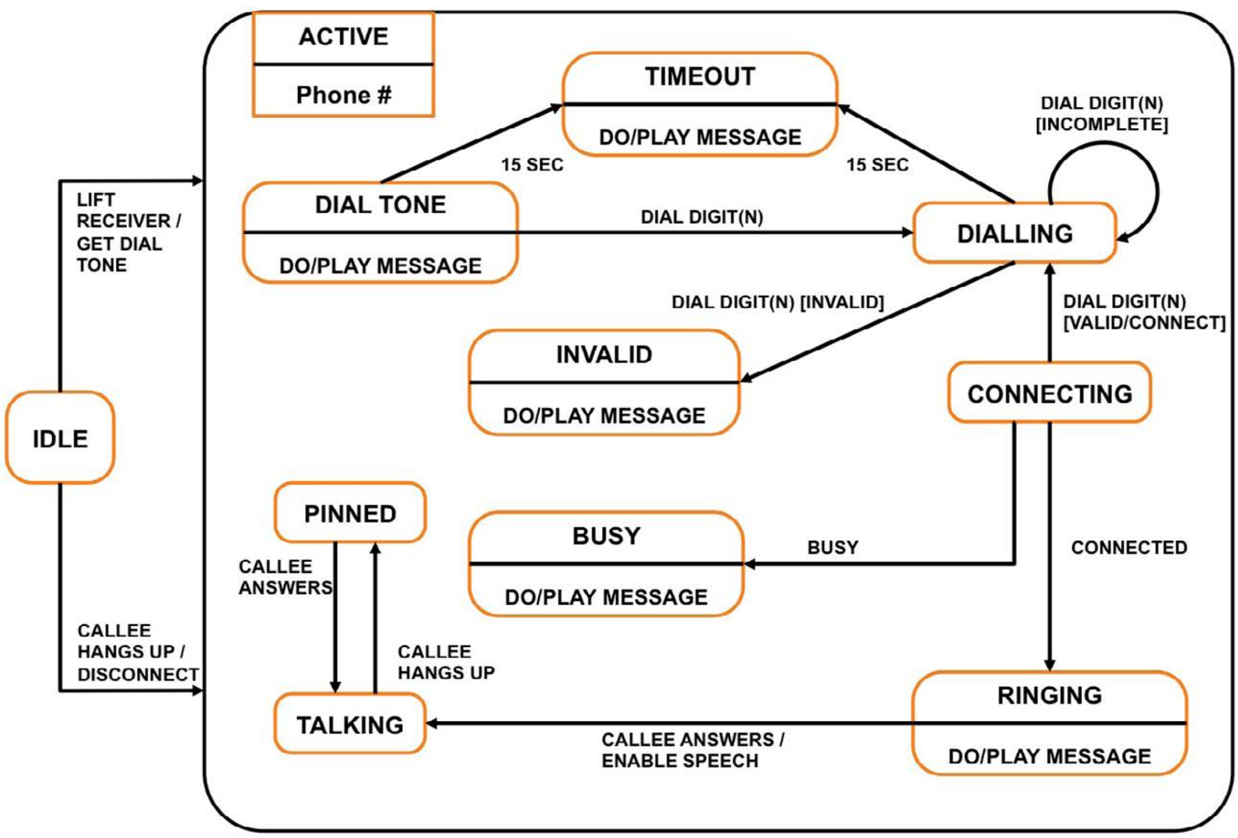
REPRESENTATION

- UML state diagram.

EXAMPLE

Example below shows a state diagram with states and state transitions.

Figure 3-33: Example P5 View



6.6 P6 – Resource Sequence

NAFv3: NSV-10C

The P6 Viewpoint is concerned with the time-ordered examination of the interactions between Resource Types.

Views implementing this Viewpoint:

- Shall specifies sequences in which data elements are exchanged in context of a Resource Type or Port.
- Shall include a time-ordered representation of the data elements exchanged between participating Resource Type or Ports.
- May represent flows of materiel, human resources or energy as interactions.

CONCERNS ADDRESSED	USAGE
• Message Handling. • Complex System Behaviours. • Security Modelling.	• Analysis of resource events impacting operation. • Behavioural analysis. • Identification of non-functional system requirements.

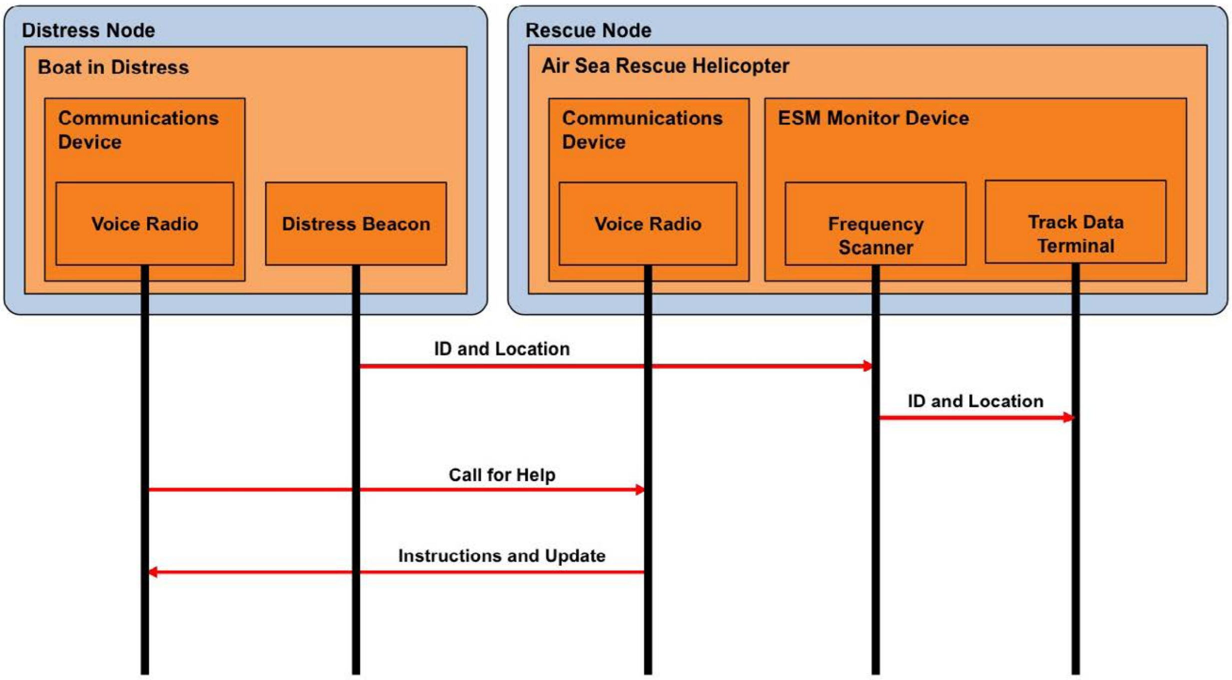
REPRESENTATION

- Topological (connected shapes).
- UML Sequence Diagram (preferred).

EXAMPLE

Figure 3-34 shows the components of an example P6 View. The items across the top of the diagram are usages of Resource Type or Ports. The lifelines are depicted as vertical lines descending from the Resource Type and Ports.

Figure 3-34: Example P6 View



6.7 P7 – Physical Data Model

NAFv3: NSV-11B

The P7 Viewpoint is concerned with the structure of data used by the resource types in the architecture.

Views implementing this Viewpoint:

- Shall map a given logical data model (L7) to the physical data model (P7) if both models are used.
- Shall describe how the information represented in the L7 Logical Data Model Viewpoint is implemented for a given solution.
- May also simply be a text schema (e.g. in the case of SQL or ISO10303-11).

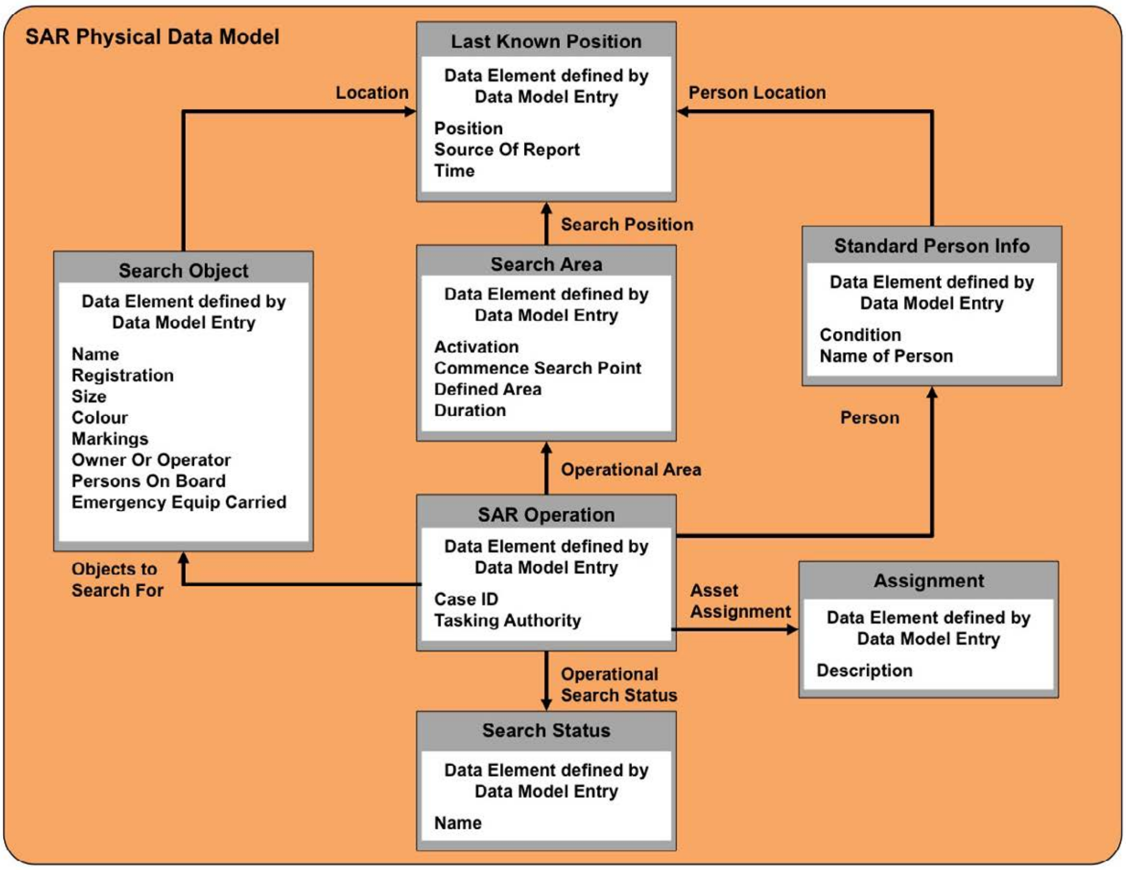
CONCERNS ADDRESSED	USAGE
• System Design. • Data Schema Design. • Message / Protocol Specification. • Data Architecture. • Database Design.	• Specifying the data elements exchanged between systems (thus reducing the risk of interoperability errors). • Definition of physical data structure (input to system design).

REPRESENTATION

- Formal text data modelling language (e.g. SQL, ISO10303-11, etc.).
- Topological (connected shapes).
- UML class diagram.

EXAMPLE

Figure 3-35: Example P7 View



6.8 P8 – Resource Constraints

NAFv3: NSV-10A

The P8 Viewpoint is concerned with functional and non-functional constraints on the implementation aspects of the architecture (i.e. the structural and behavioural elements of the Resource layer).
Views implementing this Viewpoint:

- Shall include constraints on the resource types, resource functions, data and ports.
- Shall include the rules that control, constrain or otherwise guide the implementation aspects of the architecture.

CONCERNS ADDRESSED	USAGE
• Non-Functional Requirements. • Safety Cases.	• Definition of implementation logic. • Identification of resource constraints.

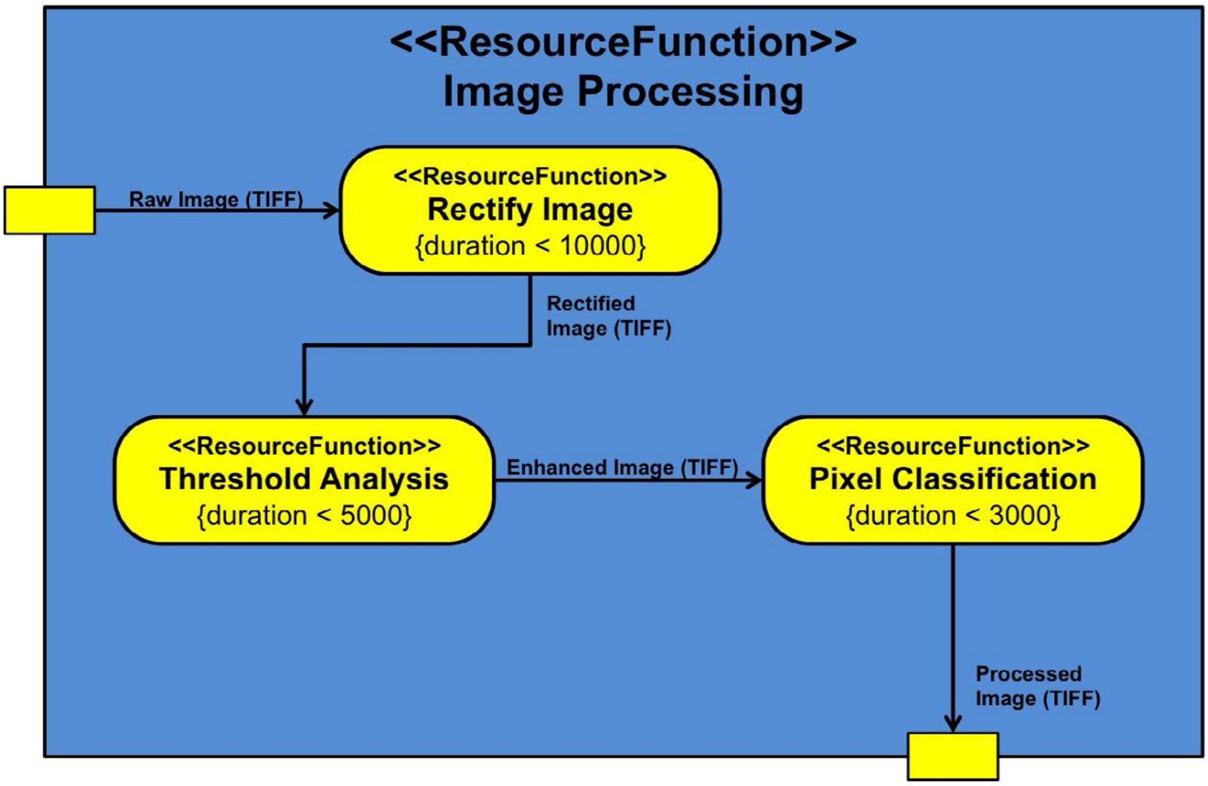
REPRESENTATION

- Text (preferably specified in a computer-interpretable constraint language such as OCL).
- Tabular.

EXAMPLE

Example below shows a set of resource functions with constraints on execution duration.

Figure 3-36: Example P8 View



6.9 Pr – Configuration Management

NAFv3: NSV-8

The Pr Viewpoint is concerned with the whole lifecycle View of a resource, describing how its configuration changes over time.

Views implementing this Viewpoint:

- Shall include an overview of how a Resource Type structure changes over time (open to all Resource Types).
- Shall include the structure of different versions of Resource Type (usually Capability Configurations or Service Implementations) mapped against a timeline.

CONCERNS ADDRESSED	USAGE
• Product Lifecycle Management. • Version Control. • Release Scheduling.	• Development of incremental acquisition strategy. • Configuration Management. • Planning technology insertion.

REPRESENTATION

- Timeline View.
- 'Herringbone' diagram.

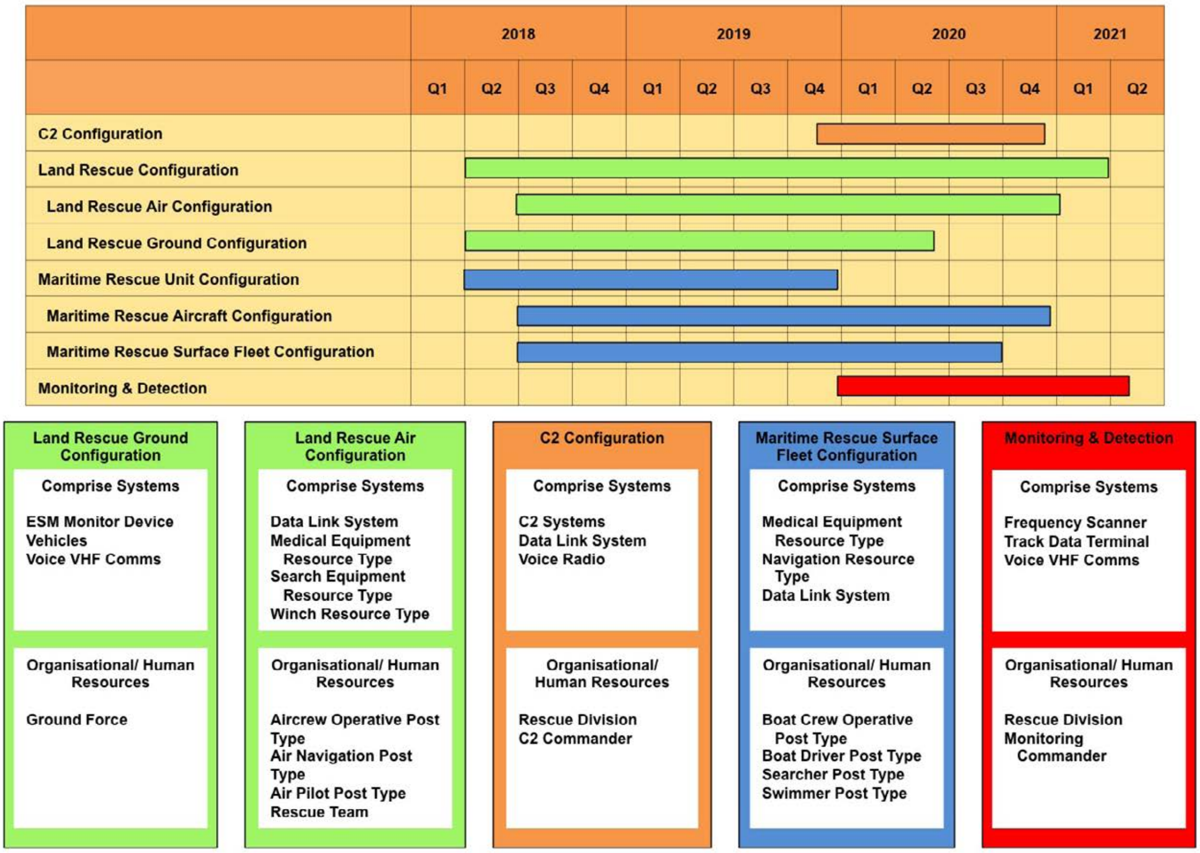
EXAMPLE

A Pr View can be used as an architecture evolution project plan or transition plan. In meta-model terms, a Pr View is constructed from data specified in the Lr, Lines of Development, and P2, Resource Structure Views, though there may be several P2 Views – one for each version of the configuration.

Using similar modelling elements as those used in the P2, Resource Structure Views, this View shows the structure of the Resource Types under configuration control. Resource interactions which take place within the Resource Type boundaries may also be shown.

The changes depicted in the Pr View are derived from the project milestones that are also shown in Lr, Lines of Development.

Figure 3-37: Example Pr View



6.10 L4-P4 – Activity to Function Mapping

NAFv3: NSV-5

The L4-P4 Viewpoint is concerned with:

- Addressing the linkage between functions described in P4, Resource Functions, and operational activities specified in L4, Logical Activities.
- Addressing the Resource Functions from the P4 Viewpoint and the Service Functions from the S4 Viewpoint.

Views implementing this Viewpoint:

- Shall depict the mapping of Resource Functions (and optionally, the resources that provide them) to operational activities or service functions.
- Shall identify the transformation of an operational need into a purposeful action performed by a system or solution.
- Shall provide the link between the services used at the operational level and the specific Resource Functions provided by the resources supporting the services.

CONCERNS ADDRESSED	USAGE
• Requirements Definition. • Process Mapping.	• Tracing functional system requirements to user requirements. • Tracing solution options to requirements. • Identification of overlaps.

REPRESENTATION

- Tabular.

EXAMPLE

Example below shows a matrix with the relationships between resource functions, and operational activities / service functions.

Figure 3-38: Example P4-L4 View

SAR Activities	Assistance Group	Locate Victim	Monitor Health	Communication Activities	Receive Distress Signal	Send Distress Signal	Monitor Group	Analyse Distress Signal	Send Warning Order	Place of Safety Activities	Process Warning Order	Recovery Activities	Maritime Rescue	Provide Medical Assistance	Recover Victim	Search Activities	Plan Search	Coordinate Search	Execute Search
Communicate																			
Broadcast Message																			
Provide Data Link													X						
Provide VHF Communications																			
Receive Message					X								X						
Send Message						X							X						
Fly Aircraft																			
Rescue Functions																			
Provide Medical Assistance													X	X					
Reassure Victim																			
Recover Victim																			
Transport Victim																			
Search Functions																			
Control Search Equipment																			X
Detect & Track Distress																			
Detect Distress Signal													X						X
Provide Track Data								X					X						X
Track Distress Signal								X					X						X

7 ARCHITECTURE META-DATA VIEWPOINTS

	Taxonomy	Structure		Connectivity	Processes	States	Sequences	Information	Constraints	Roadmap
Architecture Meta-Data	A1 Meta-Data Definitions NAV-3	A2 Architecture Products		A3 Architecture Correspondence ISO42010	A4 Methodology Used NAF Ch2	A5 Architecture Status NAV-1	A6 Architecture Versions NAV-1	A7 Architecture Meta-Data NAV-1/3	A8 Standards NTV-1/2	Ar Architecture Roadmap

Viewpoints in the Architecture Meta-Data row of the NAF grid support the administrative aspects of the architecture, such as who created it, for whom and when. Each Architecture Viewpoint is itemised and may be traced back to individual stakeholder concerns. Versions of Architecture Descriptions may be tracked and the planning (architecture roadmap) can also be captured in the Architecture Meta-Data layer.

7.1 A1 – Meta-Data Definitions

NAFv3: NAV-3

The A1 Meta-Data Definitions Viewpoint is concerned the categories of meta-data tag used throughout the architecture.
Views implementing this Viewpoint:

- Shall list all the meta-data tags used throughout the architecture.
- Shall include required meta-data tags to aid with searching and discovery.
- May list the architectural elements that are tagged, although this can be unwieldy in larger architectures.

CONCERNS ADDRESSED

- Architecture element discovery.
- Rights management.
- Protective marking at fine-grain.

USAGE

- Setting up standard tag types for the architecture.
- Assigning tags to architectural elements.

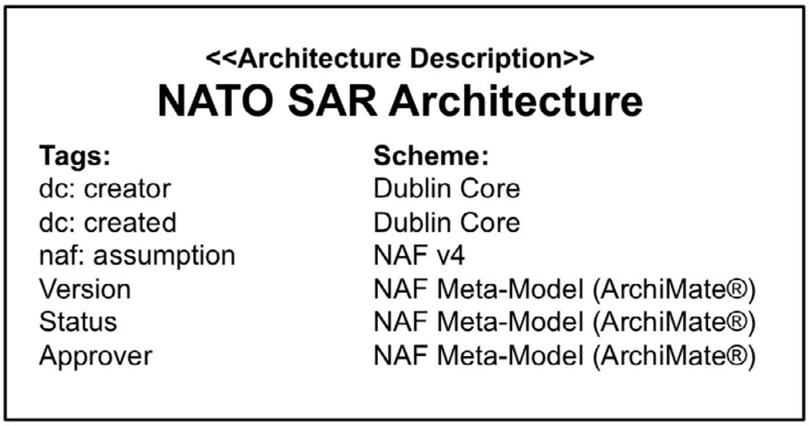
REPRESENTATION

- Tables.
- Text.

EXAMPLE

The following example A1 View simply defines the allowable tags that architects can use:

Figure 3-39: Example A1 View



The tags defined in the A1 Viewpoint can then be shown (optionally) in any other Viewpoint.
The NAF specifies some built-in meta-data tags, such as:

- Definition.
- Assumption.
- Finding.
- Recommendation.
- Purpose.

Approver, Approval Milestone, Modeler, Manager, Responsible Owner, Tool Used (see the Ar – Architecture Roadmap Viewpoint).

7.2 A2 – Architecture Products

NAFv3: NAV-3

The A2 Architecture Products Viewpoint is concerned with the products that describe an Architecture, and the Views to which those products correspond.
Views implementing this Viewpoint:

- Shall list the Views that make up the Architecture Description and which Viewpoints those Views conform to.
- Shall specify the structure of an architecture, and the products that describe the architecture.
- Shall trace the architectures onto the Enterprise Phases they correspond to (see also C2 – Enterprise Vision) and identify the key stakeholders, their concerns and the products that address those concerns (from ISO42010).
- Shall follow ISO42010, which is based on IEEE1471, as closely as possible.

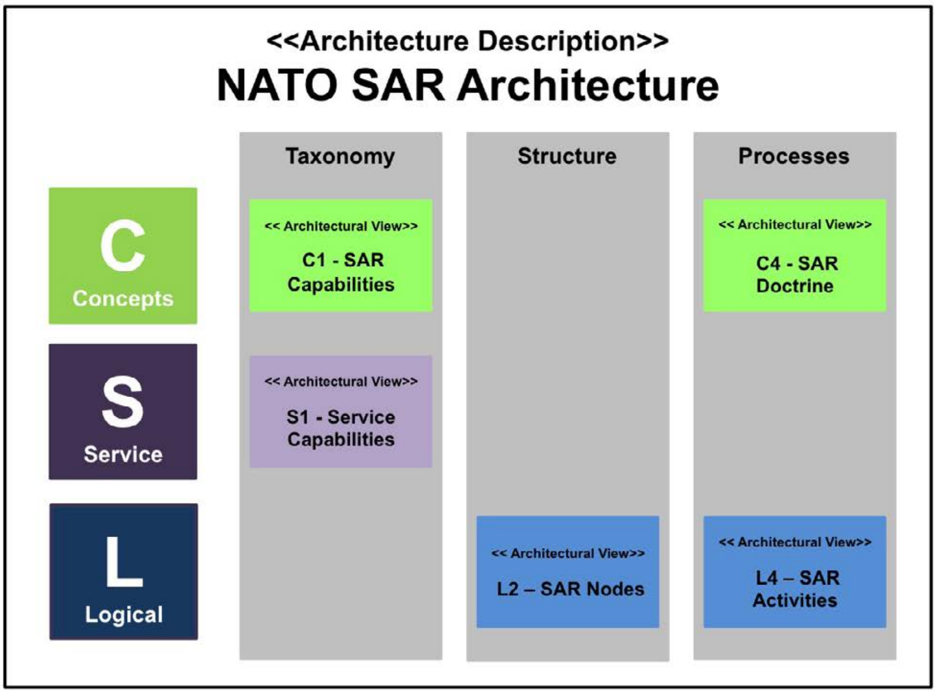
CONCERNS ADDRESSED	USAGE
• Architecture Content. • Specification and Mapping of Stakeholder Concerns.	• Summarizing an Architecture. • Navigating an Architecture.

REPRESENTATION

- NAF grid representation.
- Other representations suitable for the architect.

EXAMPLE

Figure 3-40: Example A2 View



7.3 A3 – Architecture Correspondence

NAFv3: None

The A3 Architecture Correspondence Viewpoint is concerned with the high-level dependencies between architectures.

Views implementing this Viewpoint:

- Shall include all relevant dependencies between architectures; and
- Shall implement the idea of architecture correspondence and correspondence rules complying with ISO/IEC/IEEE42010.

CONCERNS ADDRESSED

- Traceability / dependencies between architectures.
- Re-Use of Architectures.

USAGE

- Dependency analysis across architectures.

REPRESENTATION

- Tabular.
- Graphical elements linked by tracing lines.

EXAMPLE

Figure 3-41: Example A3 View

Refers To

	NATO Blogger	Armour Recovery	SPECS	NADS
NATO Blogger		X		
Armour Recovery				X
SPECS				X
NADS			X	

7.4 A4 – Methodology Used

NAFv3: None

The A4 Methodology Used Viewpoint is concerned simply state the methodology used in developing the architecture.
Views implementing this Viewpoint:
- Shall simply state the methodology used.

CONCERNS ADDRESSED	USAGE
• Architecture Management & Review.	• Architecture project management.

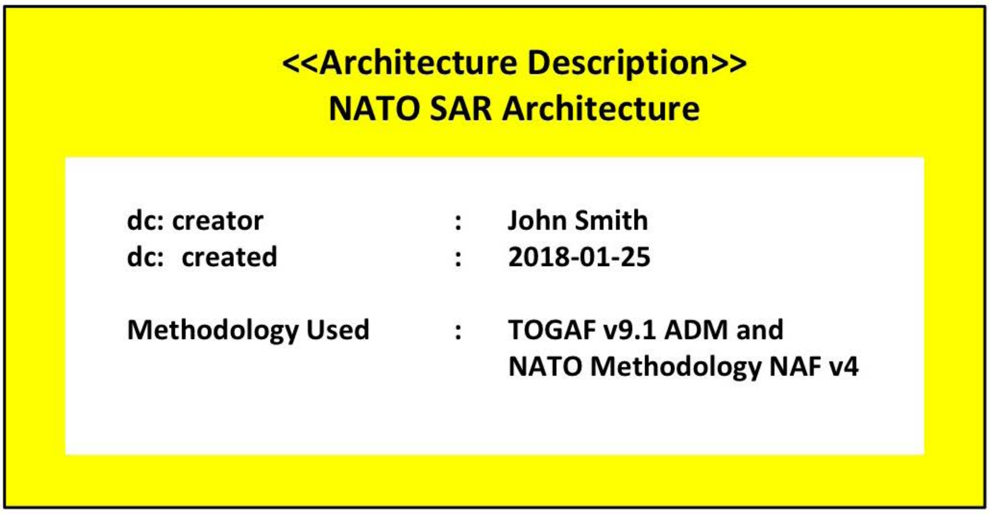
REPRESENTATION

- Tabular.
- Text Document.

EXAMPLE

As shown in the example, NAF has its own methodology, based on the TOGAF Architecture Method (ADM), specified in Chapter 2.

Figure 3-42: Example A4 View



7.5 A5 – Architecture Status

NAFv3: NAV-1

The A5 Architecture Status Viewpoint is concerned with version number and approval status of the architecture. Views implementing this Viewpoint:
- Shall assign version numbers to Views.
- Shall specify their approval dates.

CONCERNS ADDRESSED	USAGE
• Architecture Management & Review.	• Architecture project management. • Release scheduling.

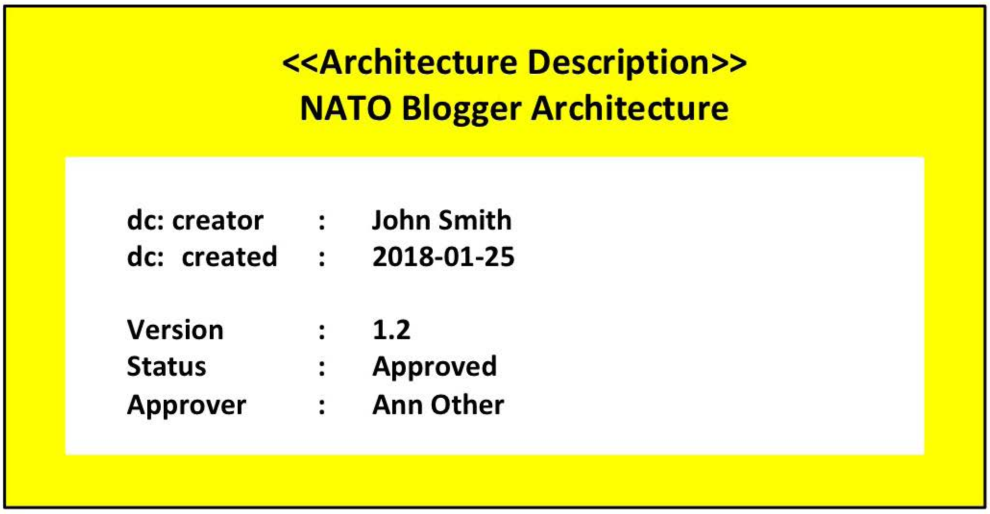
REPRESENTATION

- Tabular.
- Text Document.

EXAMPLE

The example A5 View presents a subset of the NAFv3.1 NAV-1 View. It concentrates on the versioning and status information of the architecture.

Figure 3-43: Example A5 View



Although an A5 View is usually presented as quite a simple text document or table, the underlying meta-model is much more capable, assigning architecture releases to project milestones and using succession relationships between versions. A6 and Ar Views reveal more of the underlying meta-model.

7.6 A6 – Architecture Versions

NAFv3: NAV-1

The A6 Architecture Versions Viewpoint is concerned with version history of the architecture with relevant meta-data.

Views implementing this Viewpoint:

- Shall expand on the information in the A5 by representing the catalogue of previous architecture versions.
- Shall refer to the framework used to create the architecture.
- May show relevant meta-data for each version.

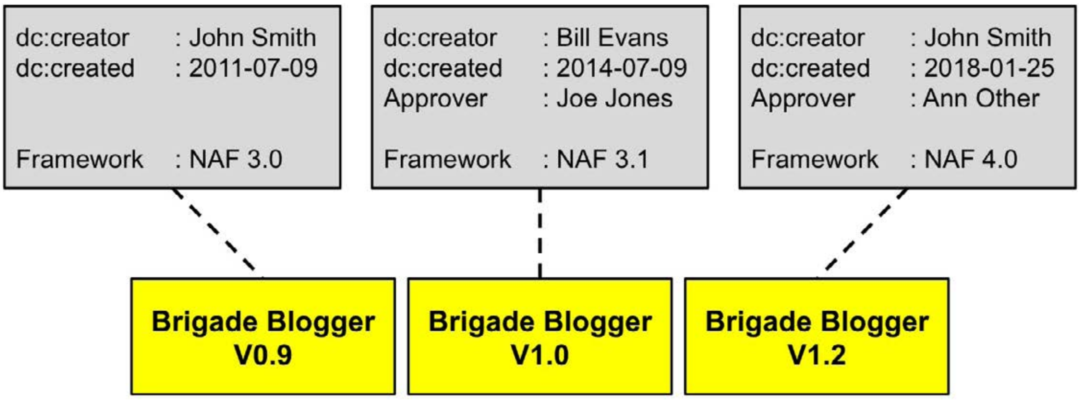
CONCERNS ADDRESSED	USAGE
• Architecture Management. • Configuration Control of Architectures.	• Recovery of old architecture models. • Version management.

REPRESENTATION

- Timeline representation.

EXAMPLE

Figure 3-44: Example A6 View



7.7 A7 – Architecture Meta-Data

NAFv3: NAV-1, 3

The A7 Architecture Meta-Data Viewpoint is concerned with the meta-data for the architecture and its Views. Views implementing this Viewpoint:

- Shall tag architecture element and Views with meta-data.
- Meta-data may include definitions, assumptions, findings, recommendations, or references.

CONCERNS ADDRESSED

- View discovery.
- Architecture discovery.

USAGE

- View tagging.
- Architecture element tagging.

REPRESENTATION

- Tabular.
- Text document.

An A7 View is usually presented as a table or text document.

EXAMPLE

The example A7 View specifies architecture meta-data tagged to the respective Views

Figure 3-45: Example A7 View

Architecture Element	Definition	Assumptions
Org A	Org A is the logistics management department.	Org A might change during implementation of this architecture.
Org B	Org B is the finance management department.	Org B will not change during implementation of this architecture.
Sys A	Sys A is the logistics management system.	Sys A will be replaced by a system delivered during implementation of this architecture.
Sys B	Sys B is the finance management system.	Sys A will not change during implementation of this architecture. All interfaces will remain the same

This differs from the A1 Viewpoint which is used to specify the types of tags that can be used throughout the architecture.

7.8 A8 – Standards	NAFv3: NTV-1, 2
The A7 Architecture Meta-Data Viewpoint is concerned with the meta-data for the architecture and its Views. Views implementing this Viewpoint: • Shall tag architecture element and Views with meta-data. • Meta-data may include definitions, assumptions, findings, recommendations, or references.	
CONCERNS ADDRESSED	**USAGE**
• Project Strategy. • Project Governance.	• Application of standards (informing project strategy). • Standards compliance. • Forecasting future changes in standards (informing project strategy). • Specifying standards that will have an impact on the architecture and the capability it is to deliver.
REPRESENTATION	
• Tabular.	

EXAMPLE

An A8 View is typically a table showing the standards used throughout the architecture. Apart from the standard itself, the table may optionally show:

- The version identifier of the standard (in accordance with AAP-03(J) for NATO standards).
- The ratification body responsible for the standard (e.g. NATO, ISO, other external military or civilian authority).
- The ratification date of the standard.
- NATO standards can be found at www.nato.int/cps/en/natolive/71191.htm.
- The publisher of the standard, if different to the ratification body (non-NATO standards only, as applicable).
- The elements in the architecture which conform to the standard.
- Any other supporting information.

The standards need not be technical, and may be related to business or military doctrine, best practice, or even legislation.

Figure 3-46: Example A8 View

Name	Version	Date	Ratification Body	Website
XMI	2.5.1	May 2015	Object Management Group	www.omg.org/spec/XMI
Archimate®	3.0.1	2007	The Open Group	http://pubs.opengroup.org/architecture/archimate3-doc/
ISO/IEC/IEEE 42010	2011		International Organization for Standardization	https://www.iso.org/standard/50508.html
NAF	4	Jan 2018	NATO	Architecture Body of Knowledge

The time from initial concept to fielded capability may be very long. It is, therefore, necessary to be able to refer to standards which, although not ratified at the time of producing the architecture, will have an impact on the capability. This could be anything from expected changes in legislation around spectrum management to future environment and safety standards. Being able to refer to emerging standards also enables the architect to mitigate the risk of outmoded specifications – so called "designed obsolescence". The A8 View may therefore also specify standards that are not currently ratified but are expected to have an impact on the fielded capability.

7.9 Ar – Architecture Roadmap

NAFv3: NAV-1

The Ar Architecture Roadmap Viewpoint is concerned with the project timeline for the architecture, including draft releases and the schedule for future releases.
Views implementing this Viewpoint:
- Shall provide detailed information regarding the architecture project from a timeline perspective.
- Shall represent the history of the architecture project as well as its future direction.

CONCERNS ADDRESSED	USAGE
• Architecture Project Management.	• Developing architectures

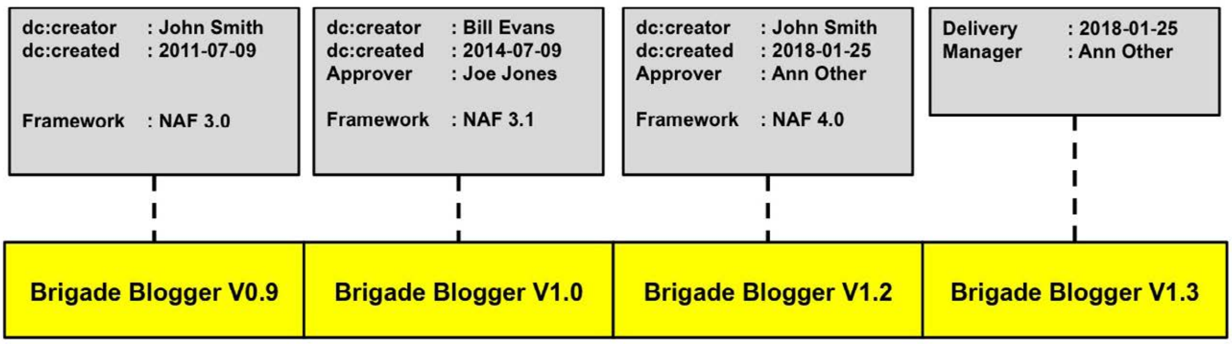
REPRESENTATION

An Ar View is usually shown as a timeline annotated with architecture releases and meta-data.

EXAMPLE

The example Ar View shows timeline relevant detail such as:
- Future architecture release schedule.
- Addition architecture milestones, such as reviews, gateways, etc.
- Links to approvals.

Figure 3-47: Example Ar View



The Ar Viewpoint provides an opportunity to display much more information than is normally shown in the A5 or A6 Viewpoint.

Chapter 4 - Meta-Model

1 INTRODUCTION

1.1 Chapter 4 of the NATO Architecture Framework identifies the meta-models to be used for creating NAFv4 compliant architectures.

1.2 ISO 42010 defines the term meta-model as something that “presents the Architecture Description (AD) elements that comprise the vocabulary of a model kind”. There are different ways of representing meta-models. The meta-model should present:

Entities: What are the major elements present in models of this kind?

Attributes: What properties do entities possess in models of this kind?

Relationships: What relations are defined among entities in models of this kind?

Constraints: What kinds of constraints are on entities, attributes and/or relationships in models of this kind?

1.3 NAFv4 compliant architectures can be creating using the following meta-models; The Open Group®’s ArchiMate® and the Object Management Group®’s Unified Architect Framework (UAF) ® Domain Meta-model (DMM)®.

2 ARCHIMATE®

2.1 ArchiMate® is an open and independent modeling language for Enterprise Architecture developed by The Open Group® to enable Enterprise Architects to describe, analyze, and visualize the relationships among architecture domains in an unambiguous way.

2.2 Although the ArchiMate® Specification does not openly call itself a framework meta-model, the document introduction states that it “offers a common language for describing the construction and operation of business processes, organizational structures, information flows, IT systems, and technical and physical infrastructure” and thus satisfies the criteria of a framework meta-model to underpin Chapter 3.

2.3 The current version of ArchiMate® can be found at https://www2.opengroup.org/ogsys/catalog/C179.

3 UNIFIED ARCHITECTURE FRAMEWORK® (UAF) DOMAIN META-MODEL (DMM)®

3.1 The Unified Architecture Framework (UAF) Domain Meta-model (DMM) is an open and non-implementation specific meta-model developed by the Object Management Group® to describe various stakeholder concerns, such as security or information, associated with a system through a set of predefined viewpoints and associated views, mapped to the corresponding view in NAFv4.

3.2 The current version of the UAF DMM can be found as Annex A to the UAF at www.omg.org/spec/UAF/Current.

Chapter 5 – Glossary, References & Bibliography

1 GLOSSARY

Term	Definition
(Architecture) Evaluation	Judgment of the value, worth, significance, importance, or quality of one or more architectures ISO/IEC 42030
Architecting	Process of conceiving, defining, expressing, documenting, communicating, assessing proper implementation of, maintaining and improving an architecture of an entity throughout its life cycle. Adapted from ISO/IEC/IEEE 42010
Architecting Principle	Declarative statement that prescribes a property of something. They reflect a level of consensus across the enterprise, and embody the spirit and thinking of the enterprise architecture. Adopted from TOGAF 9.1
Architecture	Fundamental concepts or properties of an entity of interest in its environment embodied in its elements, relationships, and in the principles of its design and evolution. Adapted from ISO/IEC/IEEE 42010
Architecture Description	Work product used to express an architecture. SOURCE: ISO/IEC/IEEE 42010
Architecture Governance	Strategic activities allowing mastering architecture according to the enterprise directions and objectives. ISO/IEC 42020
Architecture Framework	Foundational structure, or set of structures, which can be used for developing a broad range of different architectures. It should describe a method for designing a target state of the enterprise in terms of a set of building blocks, and for showing how the building blocks fit together. It should contain a set of tools and provide a common vocabulary. It should also include a list of recommended standards and compliant products that can be used to implement the building blocks. TOGAF V9.1
Architecture Principle	Declarative statement that prescribes a property of something. They reflect a level of consensus across the enterprise, and embody the spirit and thinking of the enterprise architecture. Adopted from TOGAF 9.1
Architecture Repository	Architecture Repository holds information concerning the enterprise architecture and associated artefacts. TOGAF V9.1
Architecture Style	Definition of a family of systems in terms of a pattern of structural organization. Characterization of a family of systems that are related by sharing structural and semantic properties. ISO/IEC/IEEE 24765

Architecture Vision	The Architecture Vision is created early on in the project lifecycle and provides a high-level, aspirational view of the end architecture product. The purpose of the vision is to agree at the outset what the desired outcome should be for the architecture, so that architects can then focus on the critical areas to validate feasibility. Providing an Architecture Vision also supports stakeholder communication by providing an executive summary version of the full Architecture Definition. TOGAF 9.1
Architecture View	Work product expressing the architecture from the perspective of specific concerns. Architecting outcome expressing the architecture from a given architecture viewpoint. Adapted from ISO/IEC 42010
Architecture Viewpoint	Work product establishing the conventions for the construction, interpretation and use of architecture views to frame specific concerns. Adapted from ISO/IEC/IEEE 42010
Artefact	An artefact is an architectural work product that describes an aspect of the architecture. Artefacts are generally classified as catalogues (lists of things), matrices (showing relationships between things), and diagrams (pictures of things). Examples include a requirements catalogues, business interaction matrix, and a use-case diagram. An architectural deliverable may contain many artefacts and artefacts will form the content of the Architecture Repository. TOGAF 9.1
Baseline	Agreement or result designated and fixed at a given time, from which changes require justification and approval. A specification that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development or change and that can be changed only through formal change control procedures or a type of procedure such as configuration management. ISO 24765/TOGAF V9.1
Building Block	1. An element of an entity that will be used to implement the required entity. 2. Building blocks can be defined at various levels of detail, depending on what stage of architecture development has been reached. For instance, at an early stage, a building block can simply consist of a name or an outline description. Later on, a building block may be decomposed into multiple supporting building blocks and may be accompanied by a full specification. Building blocks can relate to "architectures" or "solutions". TOGAF V9.1
Capability	A capability is the ability to achieve a desired effect under specified standards and conditions. A capability is realized through combinations of ways and means. The ability of one or more resources to deliver a specified type of effect or a specified course of action. **Note:** The term "capability" has a number of different interpretations (especially in the military community). In NAF, the term is reserved for the specification of an ability to achieve an outcome. In that sense, it is dispositional – i.e. resources may possess a Capability even if they have never manifested that capability. The MODEM definition of Capability expresses this dispositional aspect from a set-theoretic point of view; "A Dispositional Property that is the set of all things that are capable of achieving a particular outcome." Adapted from CJCSM 3170.01B

Catalogue	A structured list of architectural outputs of a similar kind, used for reference. For example, a technology standards catalogue or an application portfolio. TOGAF V9.1
Concept	An idea or mental image which corresponds to some distinct entity or class of entities, or to its essential features, or determines the application of a term (especially a predicate), and thus plays a part in the use of reason or language. Oxford Dictionary
Concern	Interest or impact in an entity relevant to one or more of its stakeholders. Note 1 to entry: When the word concern is used without any qualifier it refers to the general case. When a qualifier is prepended to the word concern, this indicates that the concern applies to the particular kind of thing, such as in the following examples: stakeholder concern, architecture concern, system concern. ISO/IEC/IEEE 42020
Configuration Management	A discipline applying technical and administrative direction and surveillance to: • Identify and document the functional and physical characteristics of a configuration item. • Control changes to those characteristics. • Record and report changes to processing and implementation status. Also, the management of the configuration of enterprise architecture practice (intellectual property) assets and baselines and the control of changeover of those assets. TOGAF V9.1
Deliverable	An work product that is contractually specified and in turn formally reviewed, agreed, and signed off by the stakeholders. Adapted from TOGAF V9.1
Driver	(Architecting / Engineering) An external or internal condition that motivates the organization to define its goals. An example of an external driver is a change in regulation or compliance rules which, for example, require changes to the way an organization operates; i.e., Sarbanes-Oxley in the US. TOGAF V9.1
DLOD	United Kingdom Ministry of Defence lines of Development.
DOTMLPFI	Doctrine, Organization, Training, Materiel, Leadership & Education, Personnel, Facilities and Interoperability/Information.
Enterprise	Project or undertaking, especially a bold or complex one. **Note:** One or more organizations will participate in an enterprise. Each of these organizations brings various resources forward for use in the enterprise and they participate to the extent that they benefit from their involvement. The purpose of the enterprise is to address some challenges that these participating organizations cannot readily address on their own. (See definition of organization below. SOURCE: Oxford English Dictionary
Enterprise Architecture	The formal description of the structure and function of the components of an enterprise, their interrelationships, and the principles and guidelines governing their design and evolution over time. MODAF V1.1
Gap	A statement of difference between two references. NATO IST-130
Goal	A high-level statement of intent or direction for an organization. Typically used to measure success of an organization. TOGAF 9.1
Lifecycle	Set of distinguishable phases or stages that an entity goes through from its conceptualization until it ceases to exist. **Note:** The architecture life cycle starts with the identification of a need for the architecture and ends with its decommissioning/discarding. The life cycle applies either to the architecture or to the architecture entity. ISO/IEC 42020

Model	A representation of a subject of interest. A model provides a smaller scale, simplified, and/or abstract representation of the subject matter. A model is constructed as a "means to an end". In the context of enterprise architecture, the subject matter is a whole or part of the enterprise and the end is the ability to construct "views" that address the concerns of particular stakeholders; i.e., their "viewpoints" in relation to the subject matter. TOGAF V9.1
Objective	An increase for an organization used to demonstrate progress towards a goal; for example, "Increase Capacity Utilization by 30% to support the planned increase in market share". Adapted for TOGAF
Organization	Group of people and facilities with an arrangement of responsibilities, authorities and relationships. ISO/IEC 42020
Pattern	A technique for putting building blocks into context; for example, to describe a re-usable solution to a problem. Building blocks are what you use: patterns can tell you how you use them, when, why, and what trade-offs you have to make in doing so. TOGAF V9.1
PESTEL	Political, Economic, Social, Technical, Environmental, Legal (Business Evaluation)
Programme	A temporary flexible organization structure created to coordinate, direct and oversee the implementation of a set of related projects and activities in order to deliver outcomes and benefits related to the organization's strategic objectives. A programme is likely to have a life that spans several years. Best Management Practice Portfolio: Common Glossary
Project	A temporary organization that is created for the purpose of delivering one or more business products according to an agreed business case. Best Management Practice Portfolio: Common Glossary
Repository	Place where work products and the associated information items are or can be stored for preservation and retrieval. ISO/IEC 42020
Requirement	A condition or capability needed by a user to solve a problem or achieve an objective. 2. a condition or capability that must be met or possessed by a system, system component, product, or service to satisfy an agreement, standard, specification, or other formally imposed documents 3. a documented representation of a condition or capability as in (1) or (2) 4. a condition or capability that must be met or possessed by a system, product, service, result, or component to satisfy a contract, standard, specification, or other formally imposed document. Requirements include the quantified and documented needs, wants, and expectations of the sponsor, customer, and other stakeholders ISO 24765
Role	The usual or expected function of an actor, or the part somebody or something plays in a particular action or event. An Actor may have a number of roles. The part an individual plays in an organization and the contribution they make through the application of their skills, knowledge, experience, and abilities. TOGAF V9.1

Solution	Result from the development, to meet the Customer, end user and the company business needs, and taking into account all applicable constraints, consisting of the System-of-Interest and its enabling systems IST-130
Stakeholder	Individual or organization having an interest in an entity or a course of action. Adapted from ISO 15288
Standard	1. Set of mandatory requirements established by consensus and maintained by a recognized body to prescribe a disciplined uniform approach or specify a product, that is, mandatory conventions and practices 2. A document that provides, for common and repeated use, rules, guidelines or characteristics for activities or their results, aimed at the achievement of the optimum degree of order in a given context ISO 24765
Strategy	1. An organization's overall plan of development, describing the effective use of resources in support of the organization in its future activities. ISO/IEC/IEEE 24765
System	A system is an integrated set of elements, subsystems, or assemblies that accomplish a defined objective. These elements include products (hardware, software, firmware), processes, people, information, techniques, facilities, services, and other support elements. Combination of interacting elements organized to achieve one or more stated purposes. **Note:** Individual System: A complete system includes all of the associated equipment, facilities, material, 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. A man-made configuration with one or more of the following: hardware, software, data, humans, processes (e.g. processes for providing service to users), procedures (e.g. operator instructions), facilities, materials and naturally occurring entities". INCOSE SE Handbook, v3.2, 2010/ISO/IEC 15288/ISO 24765
System of Interest	(Architecting) Refers to the system whose architecture is under consideration in the preparation of an architecture description. IST-130
TEPIDOIL	Training, Equipment, Personnel, Information, Doctrine, Organization, Infrastructure, and Logistics.
Traceability	A discernible association among two or more logical entities such as requirements, system elements, verifications, or tasks. SEI Glossary CMMI
Trade off Analyses	Analyses for decision-making actions that select from various requirements and alternative solutions on the basis of net benefit to the stakeholders Adopted from ISO 24765

2 STANDARDS & REFERENCE DOCUMENTS

ISO STANDARDS	
ISO/IEC 10746, 1998	Information technology. Open distributed processing. Reference model addressing information systems and Information Technologies.
ISO/IEC/IEEE 12207, 2017	Systems and software engineering – Software life cycle processes
ISO/IEC/IEEE 15288, 2015	Systems and software engineering – System life cycle processes, ISO/IEC, 2008
ISO 15704, 2000	Industrial automation systems – Requirements for enterprise-reference architectures and methodologies and close standard talking about Enterprise Modelling
ISO/DPAS 17729, draft	Unified profile for DoDAF and MODAF (UPDM)
ISO/IEC/IEEE 24765, 2017	Systems and software engineering – Vocabulary
ISO/IEC 38500, 2015	Information technology -- Governance of IT for the organisation
ISO/IEC/IEEE 42010, 2011	Systems and software engineering – Architecture description www.iso.org/iso/catalogue_detail.htm?csnumber=50508
ISO/IEC 42020 draft, 2016	Systems and software engineering — Architecture processes
ISO/IEC 42030 draft, 2016	Systems and software engineering — Architecture evaluation

INTERNATIONAL REFERENCES	
TOGAF®, 2011	TOGAF Version 9.1®, The Open Group®, 2009-2011 www.opengroup.org/architecture/togaf
FEA, 2012	Federal Enterprise Architecture, V2.3, 2012 (now obsolete), www.whitehouse.gov/omb/e-gov/fea/
ATAM, 2000	ATAM: Method for Architecture Evaluation, Rick Kazman, Mark Klein, Paul Clements, August 2000 Technical report, CMU/SEI-2000-TR-004, ESC-TR-2000-004

META-MODEL REFERENCES	
ArchiMate® 2017	ArchiMate Version 3.0.1(R), The Open Group®, 2007 www.opengroup.org/archimate/downloads
UAF DMM®	Unified Architecture Framework Domain Meta Model®, Object Management Group® www.omg.org/spec/UAF/Current

BIBLIOGRAPHY

Abusharekh, 2010	Abusharekh, A, Gloss, L, Levis, A., "Evaluation of Service Oriented Architecture-Based Federated Architectures," Wiley Online Library (wileyonlinelibrary.com), DOI 10.1002/sys.20162, 26 January 2010
Alexander, 1964	Alexander, Christopher, Notes on the Synthesis of Form, Harvard University Press, 1964, ISBN 0-674-62751-2
Ang, 2005	Huei Wan Ang, Dave Nicholson, and Brad Mercer, "Improving the Practice of DoD Architecting with the Architecture Specification Model," The MITRE Corporation, June 2005, www.mitre.org/publications/technical-papers/improving-the-practice-of-dod-architecting-with-the-architecture-specification-model
ATAM	Kazman, Rick, Mark Klein, Mario R Barbacci, Tom Longstaff, Howard Lipson, and Jeromy Carriere. July 1998. The Architecture Trade-off Analysis Method. Software Engineering Institute, CMU/SEI-98-TR-008.
Blevins, 2010	Blevins, Terry, Fatma Dandashi, and Mary Tolbert, 2010, TOGAF ADM and DoDAF Models, The Open Group White Paper.
Broy, 2009	Automotive Architecture Framework: Towards a Holistic and Standardised System Architecture Description, An overview on description concepts, models and methods.
Chen, 2008	D. Chen, G. Doumeingts, F. Vernadat, Archtiecture for enterprise integration and interoperability: Past, present and future, appearing in Computers in Industry, 59 (2008) 647-659, Elsevier B.V.
CJCSI 3170.01H	Joint Capabilities Integration and Development System (JCIDS) and JCIDS Manual, 10 January 2012. www.dtic.mil/cjcs_directives/cdata/unlimit/3170_01.pdf
CJCSI 6212.01F	Interoperability and Supportability of Information Technology and National Security Systems, 21 March 2012. www.dtic.mil/cjcs_directives/cdata/unlimit/6212_01.pdf
Dijkstra, 1974	Dijkstra, E. W., On the role of scientific thought (1974), www.cs.utexas.edu/users/EWD/transcriptions/EWD04xx/EWD447.html
DoDI 4630.08, 2004	Procedures for Interoperability and Supportability of Information Technology (IT) and National Security Systems (NSS), June 30, 2004. www.dtic.mil/whs/directives/corres/pdf/463008p.pdf
ECPD, 1947	Engineers' Council for Professional Development. (1947). Canons of ethics for engineers
Emes, 2012	M. R. Emes, P. A. Bryant, M. K. Wilkinson, P. King, A. M. James and S. Arnold, Interpreting "systems architecting" (pages 369–395) appearing in Systems Engineering Winter 2012, Volume 15, Issue 4 Article first published online: 16 May 2012 | DOI: 10.1002/sys.21202

FEAPO, 2013	Cameron, et. al., A Common Perspective on Enterprise Architecture, Architecture & Governance Magazine, Vol 9, No. 4, 2013. http://ea.ist.psu.edu/documents/A&G_Issue9_4-FEAPOcut.pdf
Greefhorst, 2011	Danny Greefhorst and Erik Proper. Architecture Principles - The Cornerstones of Enterprise Architecture, 1st Edition, Springer, 2011
Proper, 2011	Greefhorst, D., Proper, E. "The Roles of Principles in Enterprise Architecture, http://archixl.nl/files/tear2010_principles.pdf
HFM155, 2008	The NATO Human View Handbook, NATO RTO HFM-155 Human View Workshop, January 2008
Hoffman, 2007	Martin Hoffmann, Analysis of the current State of Enterprise Architecture Evaluation Methods and Practices, Information Technology Research Institute, University of Jyväskylä, Finland https://jyx.jyu.fi/dspace/bitstream/handle/123456789/41367/Article_Analysis_of_the_Current_State_of_EA_Evaluation_Methods_and_Practices.pdf?sequence=4
Hofmeister, 2007	Christine Hofmeister et al. "A general model of software architecture design derived from five industrial approaches," The Journal of Systems and Software, 2007
Kruchten, 1995	Architectural Blueprints–The "4+1" View Model of Software Architecture, Philippe Kruchten, Paper published in IEEE Software 12 (6), November 1995, pp. 42-50
Lago, 2010	Patricia Lago, Paris Avgeriou, and Rich Hilliard. Guest editors' introduction, Software Architecture: Framing Stakeholders' Concerns, IEEE Software 27(6) (November/December 2010), pp. 20–24
Lankhorst, 2013	Enterprise Architecture at work, Marc Lankhorst, Springer (Third Edition)
Li, et. al, 2011	Performance Evaluation for Industrial Automation System Integration Based on Enterprise Architecture Standards and Application in Cotton Textile Industry, in Proceedings of 2011 International Conference on System Science, Engineering Design and Manufacturing Informatization (ICSEM 2011), IEEE pp 184 – 189.
Lattanze, 2005	Lattanze, Anthony, 2005, "The Architecture Centric Development Method," Carnegie Mellon University report CMU-ISRI-05-103.
Maier, 2009	Art of Systems Architecting, Mark W. Maier, CRC Press; 3 edition (January 6, 2009)
Martin, 2004	R. Martin, E. Robertson, J. Springer, Architecture principles for enterprise frameworks. Technical Report. Computer Science Dept., Indiana Univ., 2004. www.cs.indiana.edu/Research/techreports/TR594.
MDA 2003	Overview and guide to OMG's architecture, Model Driven Architecture, OMG, http://www.omg.org/cgi-bin/doc?omg/03-06-01

Mills, 1985	John A. Mills. "A pragmatic view of the system architect". Communications of the ACM 28(7) (1985), pp. 708–717.
MITRE, 2008	Thoughts on architecture and How to Improve the Practice (Presented to Systems Engineering Colloquium Naval Postgraduate School Monterey, California, Version 3.4), www.nps.edu/Academics/Institutes/Meyer/docs/Jan%2031%20 2008Thoughts_on_Architecting.pdf
MITRE 2014a	Systems Engineering Guide, "Approaches to Architecture Development," www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/system-architecture/approaches-to-architecture-development, MITRE Corp. 2014
MITRE 2014b	Systems Engineering Guide, "Architectural Frameworks, Models, and Views," http://www.mitre.org/publications/systems-engineering-guide/se-lifecycle-building-blocks/system-architecture/architectural-frameworks-models-and-views
Muller, 2011	System Architecting: A Business Perspective, Gerrit Muller, CRC Press (September 8, 2011)
NAS, 2013	Interim Report of a Review of the Next Generation Air Transportation System Enterprise Architecture, Software, Safety, and Human Factors Copyright © National Academy of Sciences, 2013, ISBN 978-0-309-29831-5
OMG, 2003	MDA Guide version 1.0.1, OMG/03-06-01, June 2003
OMG, 2014a	The OMG Hitchhikers Guide, v7.8, OMG/2008-09-02
OMG, 2014b	Policies and Procedures of the OMG Technical Process, Ver 3.0, pp/12-12-01`
Oxford English Dictionary	https://en.oxforddictionaries.com/
RING, 2004	"An Activity-Based Methodology for Development and Analysis of Integrated DoD Architectures," Steven J. Ring, Dave Nicholson, Jim Thilenius, The MITRE Corporation, Stanley Harris, Lockheed-Martin Corporation, March 2004 www.mitre.org/publications/technical-papers/an-activitybased-methodology-for-development-and-analysis-of-integrated-dod-architectures
SABSA, 2011	TOGAF® and SABSA® Integration, white paper by The Open Group TOGAF-SABSA Integration Working Group, October 2011

SARA	H. Obbink et al. Report on Software Architecture Review and Assessment (SARA), version 1.0. Feb. 2002. url: http://philippe.kruchten.com/architecture/SARAv1.pdf
SEI, 2009	U.S. Army Workshop on Exploring Enterprise, System of Systems, System, and Software Architectures, Mike Gagliardi, John Klein, Rob Wojcik, Bill Wood, Technical Report, CMU/SEI-2009-TR-008, The Software Engineering Institute, ESC-TR-2009-008, March 2009. www.sei.cmu.edu
TUM-IBM, 2009	White Paper of the IBM Corporation and TUM Technical Report, Manfred Broy, Mario Gleirscher, Peter Kluge, Wolfgang Krenzer, Stefano Merenda, and Doris Wild, TUM-I0915, July 2009
Vitruvius, BC	De architectura, Marcus Vitruvius Pollio (1st century BC) (Transl. Morris Hicky Morgan, 1960), The Ten Books on Architecture. Courier Dover Publications. ISBN 0-486-20645-9.
Other references can be found in the bibliography of the ISO JTC1/SC7/WG42 bibliography: • www.iso-architecture.org/42010/docs/bibliography-42010.pdf • www.iso-architecture.org/iso-archeval/Archeval-Bibliography.pdf	

Please relay any editorial mistakes and/or recommendations for improvement to this document to: Mailbox NHQC3S-C3B(Secretariat) <c3bsecretariat@hq.nato.int>.

Date	**Change Type**