



NATO Architecture Framework Version 4.1

Architecture Capability Team
Digital Policy Committee
March 2026

Document Version 2026.03



Acknowledgments for NAFv4.1 Publication

Throughout the development of version 4.1 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 Canada, France, Germany and the United Kingdom.

Special thanks goes to other Nations and Industry Partners for their support in assigning and providing resources in the architecture domain.

The NATO Architecture Framework is a substantial achievement for the Architecture Capability Team under the Digital Policy Committee. Each member of the Architecture Capability Team provided 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.



Contents

1	GENERAL	12
1.1	Purpose.....	12
1.2	Aim	12
1.3	Objectives.....	12
1.4	Scope of NAF Documentation	12
1.5	Reason for Change	13
2	WHAT IS ARCHITECTURE?	14
2.1	Description	14
2.2	Why Develop Architectures?.....	14
3	WHAT IS AN ENTERPRISE ARCHITECTURE?	15
3.1	Description	15
4	WHAT IS AN ARCHITECTURE FRAMEWORK?.....	16
4.1	Description	16
5	THE STRUCTURE OF THE NATO ARCHITECTURE FRAMEWORK (NAF).....	17
5.1	Introduction.....	17
6	PURPOSE AND SCOPE OF ARCHITECTURES AND ARCHITECTURE FRAMEWORKS.....	18
6.1	Introduction.....	18
6.2	What is the Value of an Architecture?	19
6.3	Interoperability between Architectures.....	19
7	NEW FEATURES AND IMPORTANT CHANGES IN NAFv4	21
7.1	New Features	21
7.2	Architecture Methodology.....	21
7.3	Grid Representation	21
7.4	Adoption of Industry Meta-Models.....	22
7.5	Architecture Body of Knowledge	22
1	METHODOLOGY.....	24
1.1	Foreword	24
1.2	Scope	24
1.3	Why do we need this architecting methodology?	25
2	MAIN CONCEPTS FOR ARCHITECTURE AND ARCHITECTING	26
2.1	Introduction for Architecting and Architecture	26
2.2	Architecting Scope.....	26
2.3	Stakeholder Concerns, Viewpoints and Perspectives	27
2.4	Architecture Dimensions	27
2.5	Types of Architectures.....	27
2.6	Architecting Styles.....	28
2.7	Main Architecture Processes.....	28
2.8	Architecture Governance	29

2.9	Architecture Management	29
2.10	Architecture Description	30
2.11	Architecture Evaluation.....	30
2.12	Architecture Enablers.....	30
2.13	Architecture Life Cycle.....	31
2.14	Architectures and Architecting Activities in the Enterprise	31
2.15	Architecture Framework	32
2.16	Enterprise Architecture Landscape	34
2.17	Project Architecture Landscape	34
2.18	Architecture Landscape Interactions	34
2.19	Reference Libraries.....	35
2.20	Architecture Repositories.....	36
2.21	Architecture Motivation Data	37
2.22	Architecture Policy	39
2.23	Architecture Management Plan	39
2.24	Migration Plan	40
2.25	Evaluation Report.....	40
2.26	Main Architecture Document.....	40
2.27	Architecture Dashboard	41
3	ARCHITECTING ACTIVITY	43
3.1	Architecting Stages.....	43
3.2	Method:.....	44
3.3	Architecting Dynamics.....	45
3.4	Multi-level Architecting.....	47
4	ARCHITECTING FOR THE ENTERPRISE SCOPE	49
4.1	Introduction.....	49
4.2	Overview of the Enterprise Architecting Stages	49
4.3	Enterprise Architecting Activities	50
5	ARCHITECTING IN A PROJECT	59
5.1	Overview of Project Architecting Activities.....	59
5.2	Project Architecting Activities	59
6	FOUNDATION FOR ARCHITECTING.....	70
6.2	Architecting Principles (Foundation for Best Practices)	70
6.3	Overview.....	70
6.4	Description of Sub-Processes	71
6.5	Architecture Principles in NAFv4.....	73
6.6	Architecture Capabilities	74
6.7	Recommended Patterns for Architecture and Architecting.....	74
6.8	Architecture Assets	75
1	INTRODUCTION	77

1.1	Architecture Descriptions.....	77
2	NAF GRID REPRESENTATION.....	79
2.1	Description	79
2.2	NAF Information Model.....	80
2.3	Approach	80
2.4	Graphical and Typographical Interpretation	82
3	Super Structure.....	83
3.1	Abstract Superstructure	83
3.2	Logical Active Resources	84
3.3	Logical Passive Resources.....	84
3.4	Logical Behaviour	85
3.5	Physical Active Resources.....	85
3.6	Physical Passive Resources.....	86
3.7	Physical Behaviour.....	87
4	C1 - Capability Taxonomy	88
4.1	Cut-out.....	88
4.2	NAFv4 Compliant Example	89
5	C2 - Enterprise Vision	90
5.1	Cut-out.....	90
5.2	NAFv4 Compliant Example	91
6	C3 - Capability Dependencies	92
6.1	Cut-out.....	92
6.2	NAFv4 Compliant Example	93
7	C4 - Standard Processes	94
7.1	Cut-out.....	94
7.2	Notes on the Cut-out.....	94
7.3	NAFv4 Compliant Example	95
8	C5 - Effects.....	96
8.1	Cut-out.....	96
8.2	NAFv4 Compliant Example	97
9	C7 - Performance Criteria	98
9.1	Cut-out.....	98
9.2	Notes on the Cut-out.....	98
9.3	NAFv4 Compliant Example	99
10	C8 - Planning Constraints.....	100
10.1	Cut-out.....	100
10.2	Notes on the Cut-out.....	100
10.3	NAFv4 Compliant Example	101
11	Cr - Capability Roadmap	102
11.1	Cut-out.....	102

11.2	Notes on the Cut-out.....	102
11.3	NAFv4 Compliant Example	103
11.4	Notes on the Example	103
12	S1 - Service Taxonomy	104
12.1	Cut-out.....	104
12.2	NAFv4 Compliant Example	105
13	S2 - Service Structure	106
13.1	Cut-out.....	107
13.2	NAFv4 Compliant Example	108
14	S3 - Service Interfaces	109
14.1	Cut-out.....	109
14.2	Notes on the Cut-out.....	110
14.3	NAFv4 Compliant Example	110
15	S4 - Service Functions.....	111
15.1	Cut-out.....	111
15.2	NAFv4 Compliant Example	112
16	S5 - Service States	113
16.1	Cut-out.....	114
16.2	Notes on the Cut-out.....	114
16.3	NAFv4 Compliant Example	115
17	S6 - Service Sequence.....	116
17.1	Cut-out.....	117
17.2	Notes on the Cut-out.....	117
17.3	NAFv4 Compliant Example	118
18	S7 Service Interface Parameters	119
18.1	Cut-out.....	119
18.2	Notes on the Cut-out.....	120
18.3	NAFv4 Compliant Example	120
19	S8 - Service Constraints	121
19.1	Cut-out.....	121
19.2	NAFv4 Compliant Example	122
20	Sr - Service Roadmap	123
20.1	Cut-out.....	123
20.2	NAFv4 Compliant Example	124
20.3	Notes on the Example	124
21	C1-S1 Service to Capability Mapping.....	125
21.1	Cut-out.....	125
21.2	NAFv4 Compliant Example	125
22	L1 - Logical Taxonomy	126
22.1	Cut-out.....	126

22.2	Notes on the Cut-out.....	127
22.3	NAFv4 Compliant Example	127
23	L2 - Logical Structure	128
23.1	Cut-out.....	128
23.2	Notes on the Cut-out.....	128
23.3	NAFv4 Compliant Example	129
24	L2-L3 - Logical Concept	130
24.1	Cut-out.....	130
24.2	NAFv4 Compliant Example	130
25	L3 - Logical Interactions.....	131
25.1	Cut-out.....	131
25.2	Notes on the Cut-out.....	132
25.3	NAFv4 Compliant Example	132
26	L4 - Logical Activities	133
26.1	Cut-out.....	134
26.2	NAFv4 Compliant Example	135
26.3	Notes on the Example	135
27	L5 - Logical States.....	136
27.1	Cut-out.....	137
27.2	NAFv4 Compliant Example	138
28	L6 Logical Sequence	139
28.1	Cut-out.....	139
28.2	NAFv4 Compliant Example	140
29	L7 - Information Model	141
29.1	Cut-out.....	141
29.2	NAFv4 Compliant Example	142
30	L8 - Logical Constraints.....	143
30.1	Cut-out.....	143
30.2	NAFv4 Compliant Example	144
31	Lr - Logical Roadmap	145
31.1	Cut-out.....	145
31.2	NAFv4 Compliant Example	146
31.3	Notes on the Example	146
32	P1 - Resource Taxonomy	147
32.1	Cut-out.....	148
32.2	Notes on the Cut-out.....	149
32.3	NAFv4 Compliant Example	149
33	P2 - Resource Structure	150
33.1	Cut-out.....	151
33.2	Notes on the Cut-out.....	151

33.3	NAFv4 Compliant Example	152
34	P3 - Resource Interactions	153
34.1	Cut-out.....	154
34.2	NAFv4 Compliant Example	155
35	P4 - Resource Functions	156
35.1	Cut-out.....	157
35.2	NAFv4 Compliant Example	158
36	L4-P4 - Activity to Function Mapping	159
36.1	Cut-out.....	159
36.2	NAFv4 Compliant Example	160
37	P5 - Resource States.....	161
37.1	Cut-out.....	161
37.2	NAFv4 Compliant Example	162
38	P6 Resource Sequence	163
38.1	Cut-out.....	163
38.2	NAFv4 Compliant Example	164
39	P7 - Data Model	165
39.1	Cut-out.....	166
39.2	NAFv4 Compliant Example	167
40	P8 - Resource Constraints.....	168
40.1	Cut-out.....	168
40.2	NAFv4 Compliant Example	169
40.3	Notes on the Example	169
41	Pr - Resource Roadmap	170
41.1	Cut-out.....	170
41.2	NAFv4 Compliant Example	171
41.3	Notes on the Example	171
42	A1 - Metadata Definitions	172
42.1	Cut-out.....	172
42.2	NAFv4 Compliant Example	173
43	A2 - Architecture Products	174
43.1	Cut-out.....	174
43.2	NAFv4 Compliant Example	175
44	A3 - Architecture Correspondence	176
44.1	Cut-out.....	176
44.2	Notes on the Cut-out.....	176
44.3	NAFv4 Compliant Example	177
45	A4 - Architecture Methodology	178
45.1	Cut-out.....	178
45.2	NAFv4 Compliant Example	179

46	A5 - Architecture Status	180
46.1	Cut-out.....	180
46.2	Notes on the Cut-out.....	180
46.3	NAFv4 Compliant Example	181
47	A6 - Architecture Versions	182
47.1	Cut-out.....	182
47.2	Notes on the Cut-out.....	182
47.3	NAFv4 Compliant Example	183
48	A7 - Architecture Metadata	184
48.1	Cut-out.....	184
48.2	Notes on the Cut-out.....	185
48.3	NAFv4 Compliant Example	185
49	A8 - Architecture Standards	186
49.1	Cut-out.....	187
49.2	Notes on the Cut-out.....	187
49.3	NAFv4 Compliant Example	188
50	Ar - Architecture Roadmap	189
50.1	Cut-out.....	190
50.2	NAFv4 Compliant Example	191
50.3	Notes on Example.....	191
1	METAMODEL	193
1.1	Introduction.....	193
1.2	ARCHIMATE®	193
1.3	UNIFIED ARCHITECTURE FRAMEWORK® (UAF) DOMAIN META-MODEL (DMM)®	193
1	Glossary	196
2	Standards & Reference Documents	203
3	Bibliography	204
4	Document Change	207



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 strategies. Purpose of Architecting is to support decision makers by providing a coherent and detailed view to satisfy analysis needs.
- 1.1.2 Architecting applies principles and practices to guide organizations through the business/mission, information, application and technology changes necessary to implement their strategies¹.
- 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 NAF viewpoints, and the relationship with the commercial meta-model constructs.

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

1.5 Reason for Change

- 1.5.1 NAF version 3 (NAFv3) was issued in 2007² 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,
 - 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/IEEE 42020 Systems and Software Engineering – Architecture Processes,
 - ISO/IEC/IEEE 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 NAFv3 was issued as Annex 1 to AC/322-D(2007)0048, was released to the public with AC/322-D(2015)0009. It replaced MODAF Version 1.2.00

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, compared³, 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.

³Note: 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 analyzing 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 levels⁴:

- **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 Program 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 threads⁵ and scenarios⁶, thus providing an environment for thorough testing of identified use cases.⁷

6.3 Interoperability between Architectures

6.3.1 Architectures must not be produced for the sake of architectures themselves, but as a means

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

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.

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:

	Taxonomy		Structure		Behaviour				Information	Constraints	Roadmap
	Connectivity	Processes	States	Sequences							
Concepts	C1 Capability Taxonomy	C2 Enterprise Vision	C3 Capability Dependencies	C4 Standard Processes	C5 Effects			C7 Performance Criteria	C8 Planning Constraints	Cr Capability Roadmap	
	C1-S1										
Service Specifications	S1 Service Taxonomy	S2 Service Structure	S3 Service Interfaces	S4 Service Functions	S5 Service States	S6 Service Sequence	S7 Service Interface Parameters	S8 Service Constraints	Sr Service Roadmap		
Logical Specifications	L1 Logical Taxonomy	L2 Logical Structure	L3 Logical Interactions	L4 Logical Activities	L5 Logical States	L6 Logical Sequence	L7 Information Model	L8 Logical Constraints	Lr Logical Roadmap		
				L4-P4							
Physical Resource Specifications	P1 Resource Taxonomy	P2 Resource Structure	P3 Resource Interactions	P4 Resources Functions	P5 Resources States	P6 Resource Sequence	P7 Data Model	P8 Resource Constraints	Pr Resource Roadmap		
Architecture Foundation	A1 Metadata Definitions	A2 Architecture Products	A3 Architecture Correspondance	A4 Architecture Methodology	A5 Architecture Status	A6 Architecture Versions	A7 Architecture Metadata	A8 Architecture Standards	Ar Architecture Roadmap		

1. 1 - NAF Viewpoints

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 is part of the NATO Architecture Body of Knowledge. The Body of Knowledge includes 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 METHODOLOGY

1.1 Foreword

- 1.1.1 The NATO Architecture Framework version 4 (NAFv4) is a standard for developing architectures.
- 1.1.2 The purpose of this Chapter is to provide a NAFv4 methodology to set up an architecting environment, governing, managing, defining, evaluating and using architectures.
- 1.1.3 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.

1.2 Scope

- 1.2.1 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.
- 1.2.2 This methodology and the formalism described in Chapters 3 and 4 are to be considered as a constructive generic framework.
- 1.2.3 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.
- 1.2.4 Chapter 5 of the NAF includes a glossary that provides specific definitions of terms used in this chapter.
- 1.2.5 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 program/project level** covers libraries, repositories, portfolios and activities used in a capability program 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.
- 1.2.6 These are illustrated in 2. 2

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
- Architecture Body of Knowledge (ABoK)

Note:

Capabilities Governance with the whole enterprise scope.

Capability management per project.

Artefact description addressed by Enterprise and project

2. 2 - Three Main Methodological Areas

1.3 Why do we need this architecting methodology?

- 1.3.1 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.
- 1.3.2 This methodology provides a foundation to set-up architecting activities within an organization with necessary and justified tailoring to fit with particular architecting context.
- 1.3.3 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.

2 MAIN CONCEPTS FOR ARCHITECTURE AND ARCHITECTING

2.1 Introduction for Architecting and Architecture

- 2.1.1 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.
- 2.1.2 The subject of interest may be anything, including a collection of things, analyzed 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.
- 2.1.3 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:
- 2.1.4 the scope of the architecting effort,
- stakeholder concerns, and
 - architecting activities to include producing an architecture description.
- 2.1.5 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.
- 2.1.6 The architecture of an entity, as defined by ISO/IEC/IEEE 42020⁸, 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 DLOD⁹ PESTEL¹⁰, DOTPMLFI¹¹, etc.).

2.2 Architecting Scope

- 2.2.1 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.
- 2.2.2 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.

⁸ISO/IEC/IEEE 42020 Enterprise, Systems and software — Architecture Processes

⁹DLOD: United Kingdom Ministry of Defense Lines of Development

¹⁰PESTEL: Political, Economic, Social, Technical, Environmental, Legal (Business Evaluation)

¹¹ OTMPLFI: 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

- 2.2.3 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 aesthetic¹² concerns that influence the solution.

2.3 Stakeholder Concerns, Viewpoints and Perspectives

- 2.3.1 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.
- 2.3.2 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, 1985].
- 2.3.3 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.

2.4 Architecture Dimensions

- 2.4.1 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.
- 2.4.2 Architecture viewpoints and perspectives can also be considered as dimensions that transverse the previous ones.

2.5 Types of Architectures

- 2.5.1 The NAF methodology is independent of the various types of architectures and architecting styles currently used in industry and governmental organizations.
- 2.5.2 Nevertheless, different types 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, the development and/or the application of architectures of several types. For example, an organization might define types of architectures as:
- enterprise-wide architecture describing the future situation with limited detail. This description normally covers several programs,
 - architecture description to be used as reference by a capability/program or for architecting within a domain, and

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

- 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 type 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 types of architectures are also defined in the the NATO Enterprise Architecture Policy adapting The Open Group Architecture Framework (TOGAF):

Table 2. 1 Architecture Types defined by NATO EA Policy

Architecture Types	Usages
Business Architecture	Describing the business strategy, management, organization, and key business processes (including process ownership and key decisions) of the organization.
Information Architecture	Describing the structure of an organization’s logical and physical information assets and the associated data management resources and linking the information required to the key business processes and decisions.
Application Architecture	Providing a blueprint for the individual application systems to be deployed, the information which they provide, the interactions between the application systems and their relationships to the core business processes of the organization with the frameworks for services to be exposed as business functions for integration.
Technology Architecture	Describing the hardware, software and network infrastructure needed to support the development of the application systems.

2.6 Architecting Styles

- 2.6.1 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.
- 2.6.2 Four styles of architecting have been identified by architecture practitioners within the United Kingdom (see Evans, 2013 and Evans, 2018). They are as follows:
- authoritative,
 - directive,
 - coordinative, and
 - supportive.

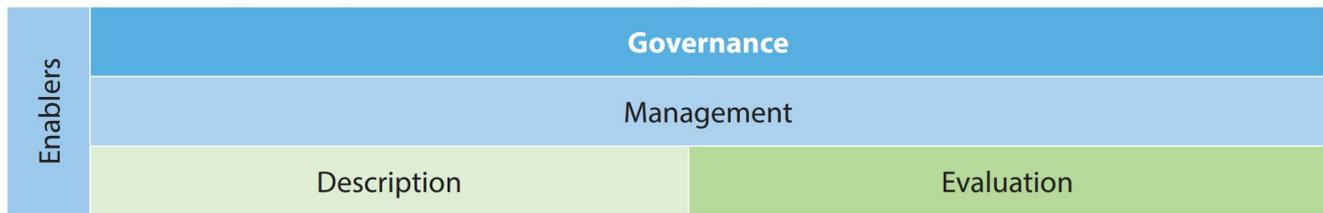
2.7 Main Architecture Processes

- 2.7.1 A first description of process, activities and tasks related to Architecture definition is provided by ISO/IEC/IEEE 15288¹³. A more detailed explanation is given in this section with identification

¹³ISO/IEC/IEEE 15288 Systems and Software Engineering — System Life Cycle Processes

of 5 processes that could be performed by different organizations and projects within an Enterprise.

2.7.2 This description of processes is close to the ISO/IEC/IEEE 42020¹⁴.



2.3 - Architecture Processes

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

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

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

2.8 Architecture Governance

2.8.1 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.

2.8.2 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.

2.9 Architecture Management

2.9.1 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.

2.9.2 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,
- monitoring and assess architecture development with management direction, and
- deciding on necessary corrective actions and iterate.

¹⁴ISO/IEC/IEEE 42020 2016 Enterprise, Systems and Software — Architecture Processes

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

2.10 Architecture Description

2.10.1 Architecture description process aims to be compliant to ISO/IEC/IEEE 42010¹⁵. The main activities identified are:

- analyze 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.

2.11 Architecture Evaluation

2.11.1 Architecture Evaluation process aims to be compliant to ISO/IEC/IEEE 42020 and ISO/IEC/IEEE 42030¹⁶. 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.

2.12 Architecture Enablers

2.12.1 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.

2.12.2 **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.

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

¹⁶ISO 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.

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

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

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

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

2.13 Architecture Life Cycle

2.13.1 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.

2.13.2 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.

2.13.3 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 program. The architecture life cycle starts with the beginning of the program 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.

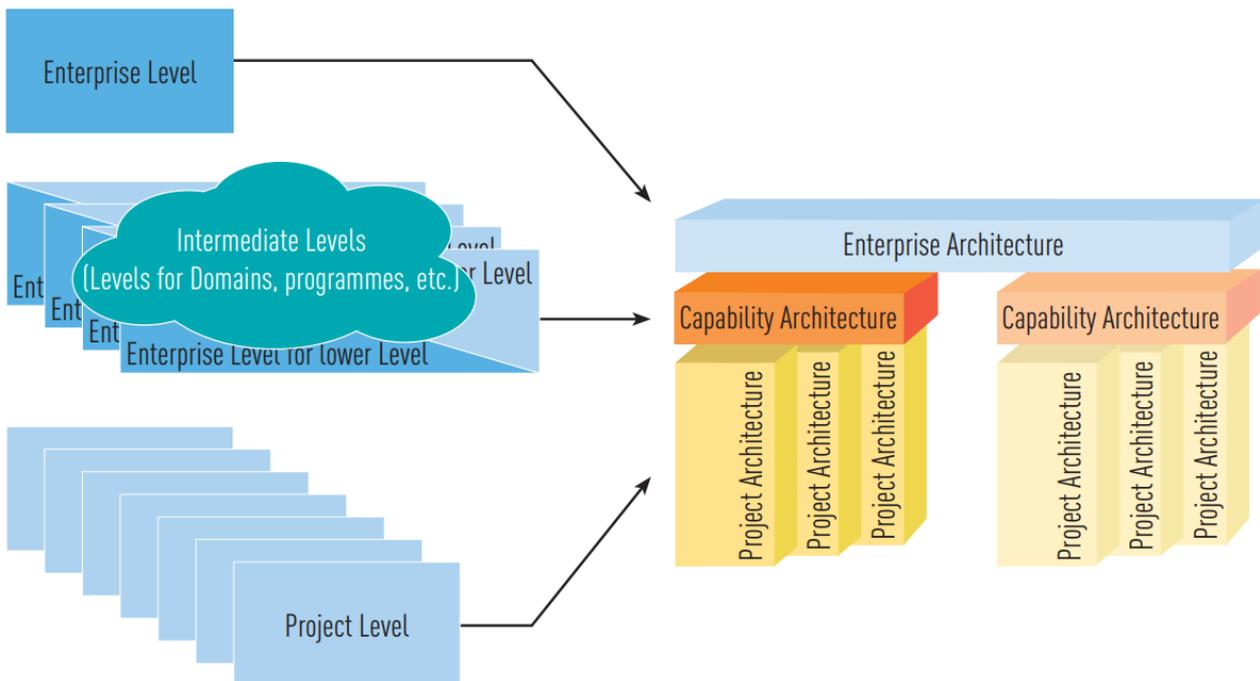
2.13.4 These examples highlight the need to customize architectural environments, activities and outcomes in order to be fit for purpose. Customization will also depend on the enterprise organization and the complexity of both problem and solution, which can call for different plans and activities on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

2.14 Architectures and Architecting Activities in the Enterprise

2.14.1 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 expertise.

- 2.14.2 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.
- 2.14.3 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.
- 2.14.4 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-level (or multi-layer) sequential activities, multi-level concurrent activities, etc.
- 2.14.5 For multi-level architecting activities in an enterprise, the middle-levels act as Project for the upper level and as Enterprise for the lower level. 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.
- 2.14.6 The following figure provides an example about how to map the multi-level architecture activities with the examples of different types of architecture.



2. 4 - Example of Multi-Level Architecture Activities

- Enterprise Architectures are developed by the enterprise level activities,
- Capability Architectures are developed by domain and program level activities,
- Project Architectures are developed at project levels.

2.15 Architecture Framework

2.15.1 Architecture Framework TOGAF v9.1, page 45:

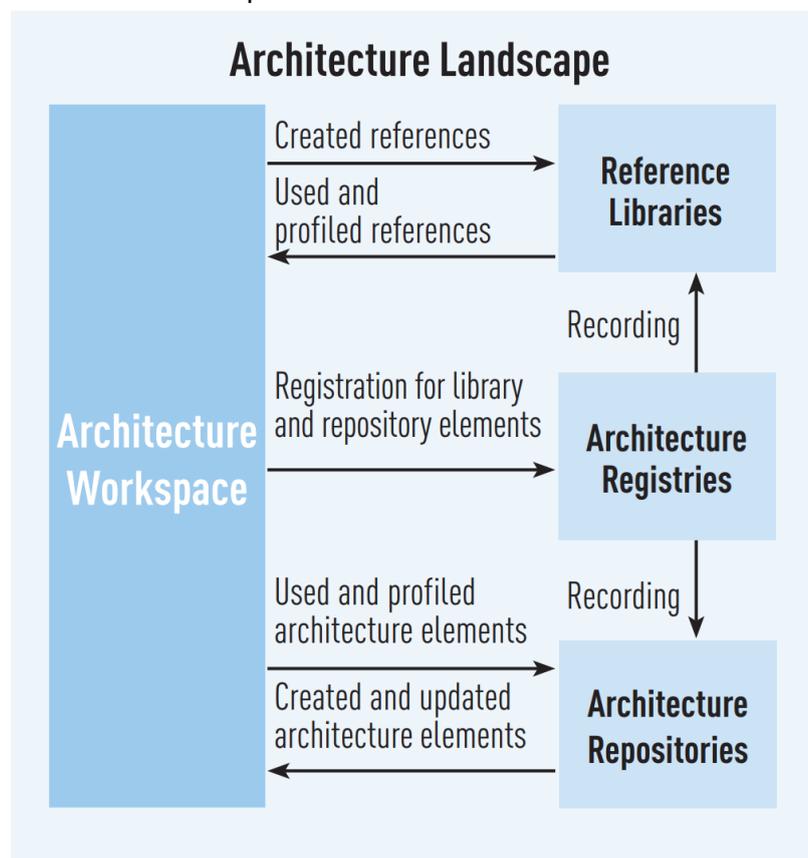
“is a foundational structure, or set of structures, which can be used for developing a broad range of different

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.

2.15.2 An architecture framework should be used as a working environment. This environment is called an ‘architecture landscape’.



2. 5 - Architecture Landscape

2.15.3 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,

2.15.4 Architecture landscapes can be considered at any level of the Enterprise performing architecting activities or accessing architecting outcomes: whole enterprise, domains,

programs and projects.



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

2.16 Enterprise Architecture Landscape

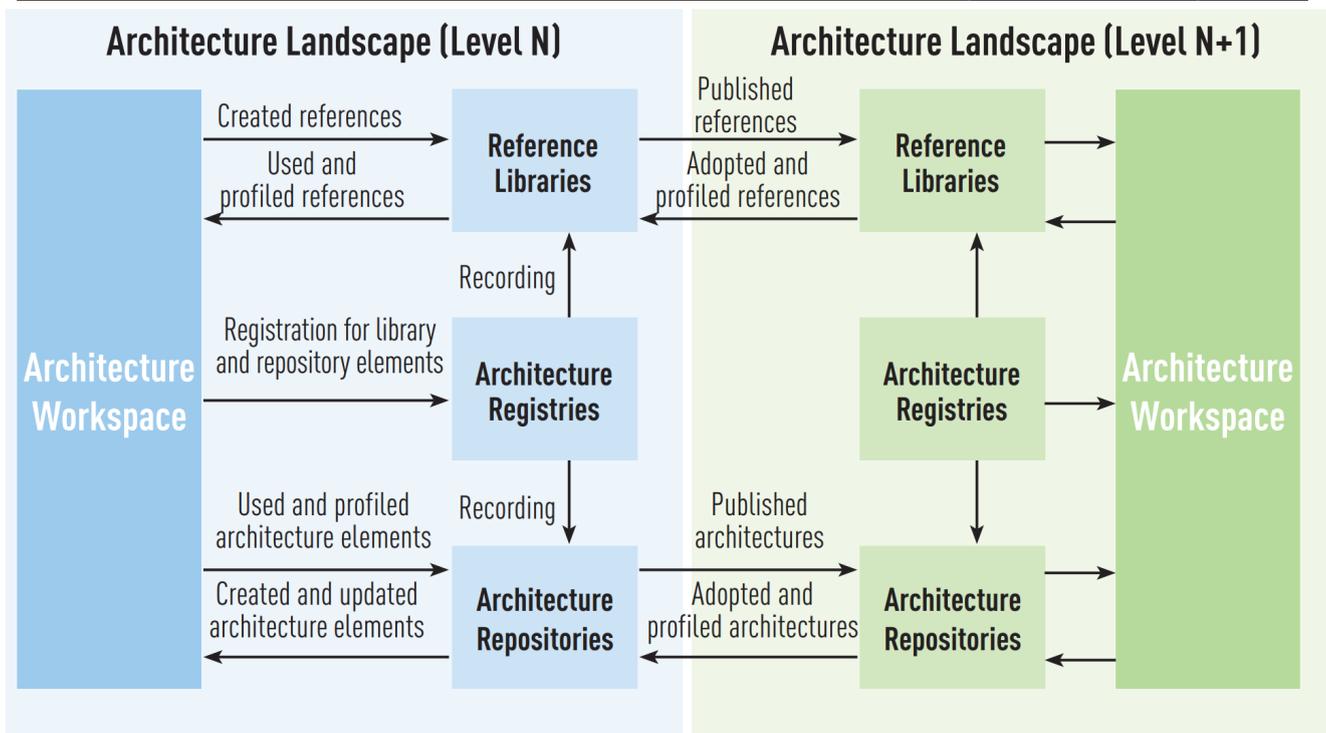
- 2.16.1 It allows enterprise architecting activities in the enterprise to cover multi-program, multi-project and enterprise-width business,
- 2.16.2 Enterprise reference libraries and Enterprise Architecture repository host data being available for the other stakeholders of the enterprise. In these 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,
- 2.16.3 Enterprise reference libraries host the baselines of assets reusable by any architect of the enterprise,
- 2.16.4 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,
- 2.16.5 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,
- 2.16.6 Enterprise Architecture registries record the usage of elements of reference libraries and of architecture repositories in the Enterprise Architecture landscape.

2.17 Project Architecture Landscape

- 2.17.1 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.

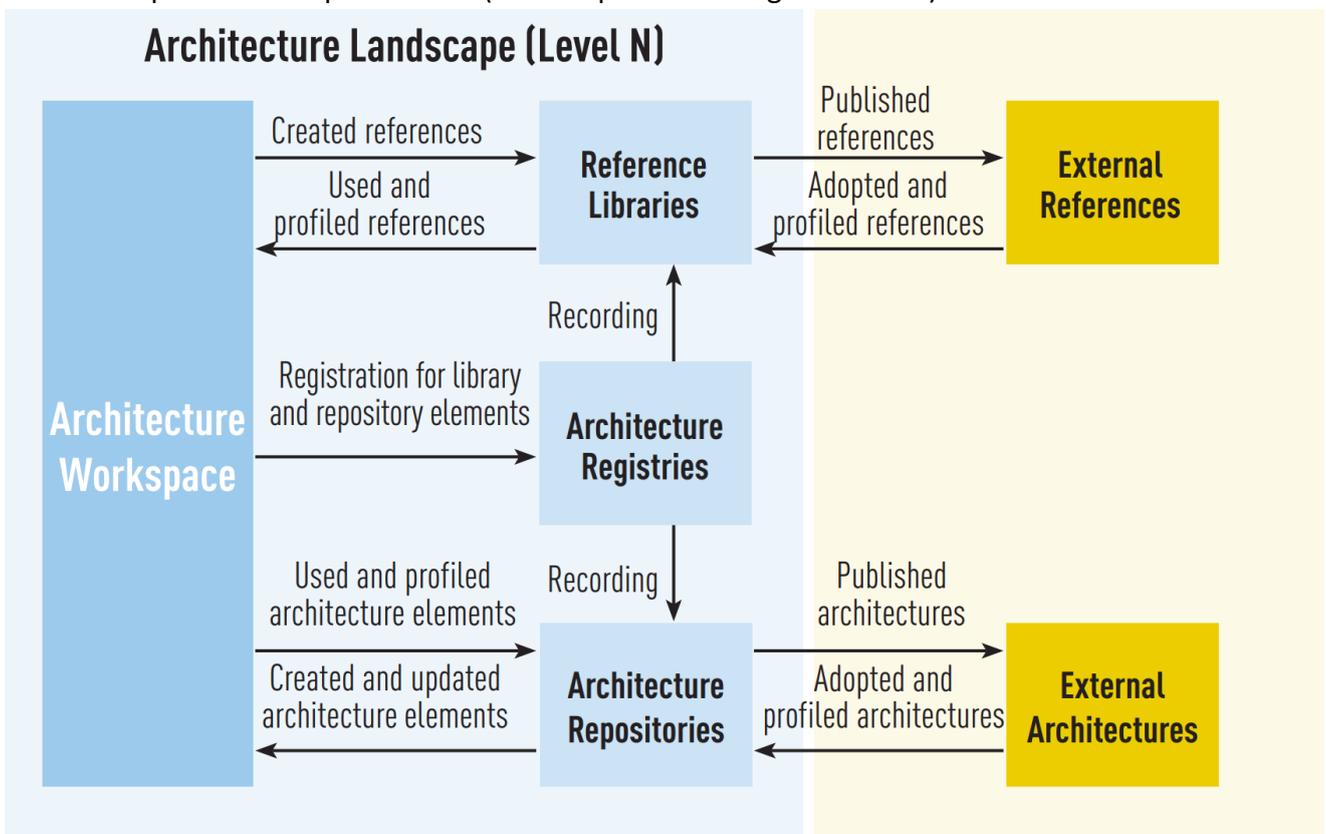
2.18 Architecture Landscape Interactions

- 2.18.1 Interactions occur between architecture landscapes when multi-level architecting activities are in place in an enterprise (See 2. 6 with the example of enterprise, domains, programs and projects levels). Architecture landscapes are complementary structures. Considering the interaction from one levels point of view:
 - the architecture landscape exposes usable or mandatory data (references and architectures) for the other levels, and
 - the architecture landscape uses and profiles data elaborated by the other levels.
- 2.18.2 Architecture landscapes also interact with the enterprise environment to:



2. 6 - Architecture Landscape Interactions (view from Level N)

- 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).



2. 7 - Architecture Landscape External Interactions

2.19 Reference Libraries

2.19.1 Reference libraries host the baselines of assets reusable by architects in their activities per architecting organization. This information can:

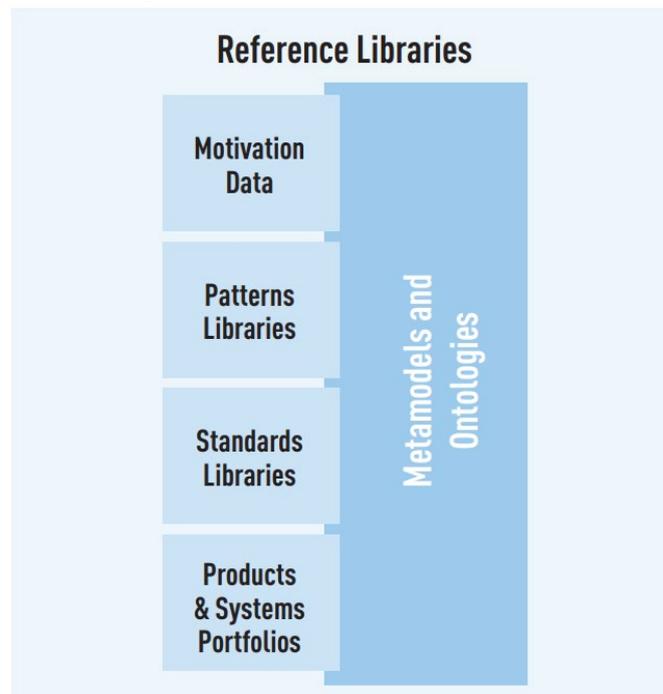
- either come from the lower architecting levels in the enterprise organization, in which case

the consistency and the relevance for the current levels is checked, or

- be created and/or collected for lower levels through architecting activities.

2.19.2 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 2. 10) 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.



2. 8 - Reference Libraries

- 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.



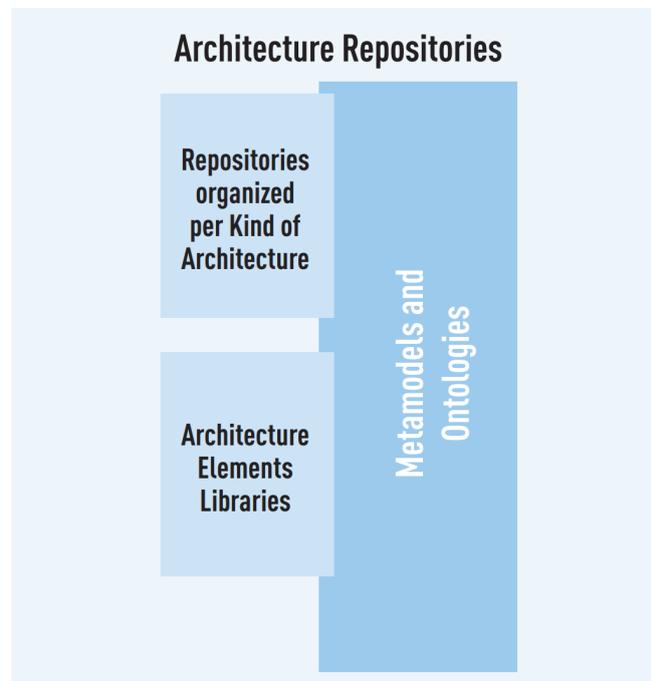
A more detailed description of architecture motivation data is given in “Architecture Repositories”, the next section.

2.20 Architecture Repositories

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

2.20.2 Architecture repositories include:

- the different types 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.

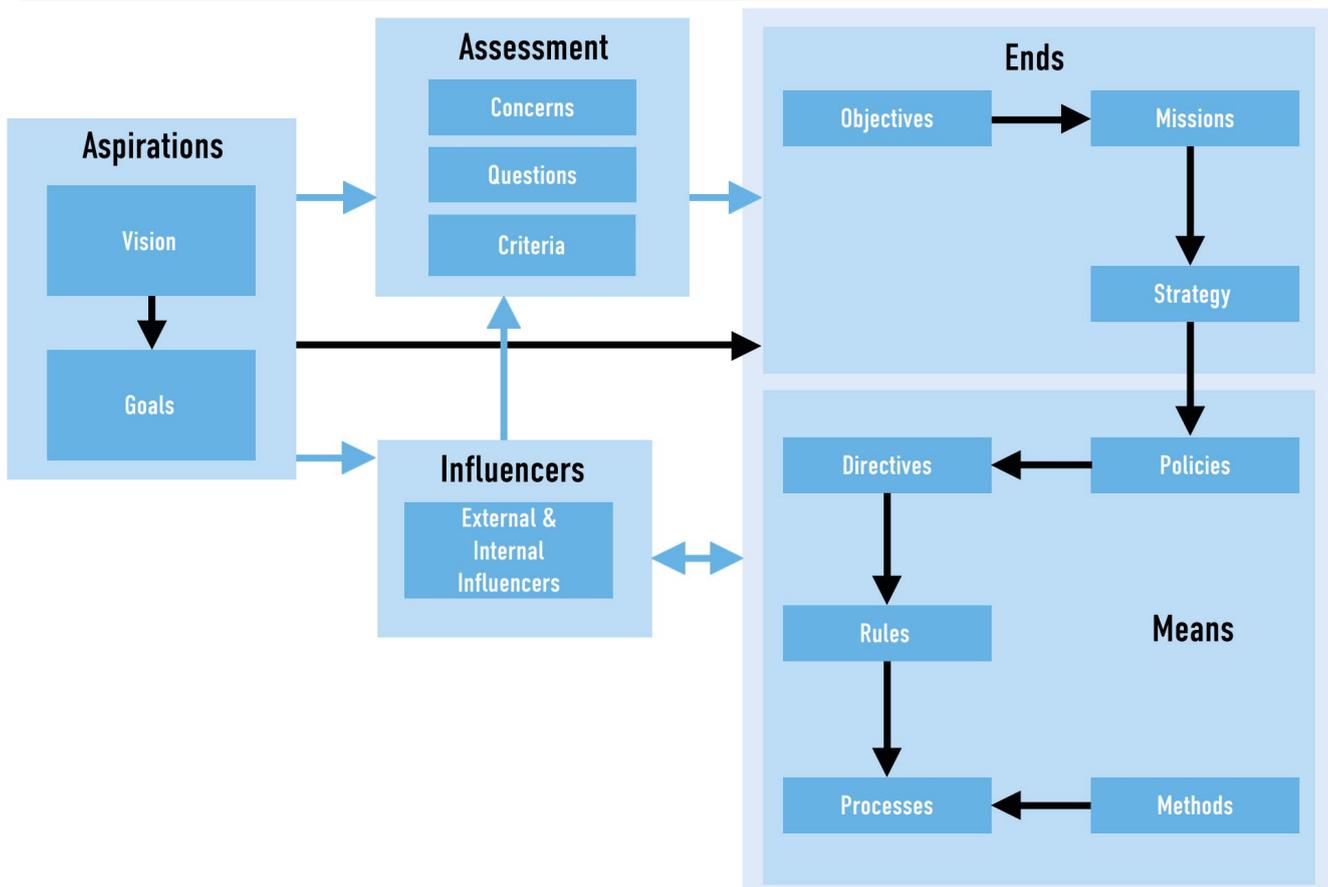


2. 9 - Architecture Repositories

2.21 Architecture Motivation Data

- 2.21.1 Architecture motivation data gathers information and references relevance for initialization of architecture, orientation of architecting activities and analysis of findings.
- 2.21.2 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.
- 2.21.3 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.
- 2.21.4 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.
- 2.21.5 Policies and rules set the context of process adaptation to major architecture drivers such as interoperability.
- 2.21.6 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.
- 2.21.7 The Business Motivation Model Version 1¹⁷ defines the relationships between various motivation elements. These relationships are shown at 2. 10.

¹⁷ OMG Document: formal/2008-08-02



2. 10 - Motivation Data

2.21.8 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 DOTMLPFI) as agreed by stakeholders).

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

- Evolution 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.

2.21.10 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.

2.22 Architecture Policy

2.22.1 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.

2.22.2 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.

2.22.3 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.

2.22.4 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.

2.23 Architecture Management Plan

2.23.1 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 2.16 to 2.18:
- **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.

2.23.2 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.

2.24 Migration Plan

2.24.1 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.

2.25 Evaluation Report

2.25.1 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.

2.25.2 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.

2.26 Main Architecture Document

2.26.1 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.

2.26.2 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.

2.26.3 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).

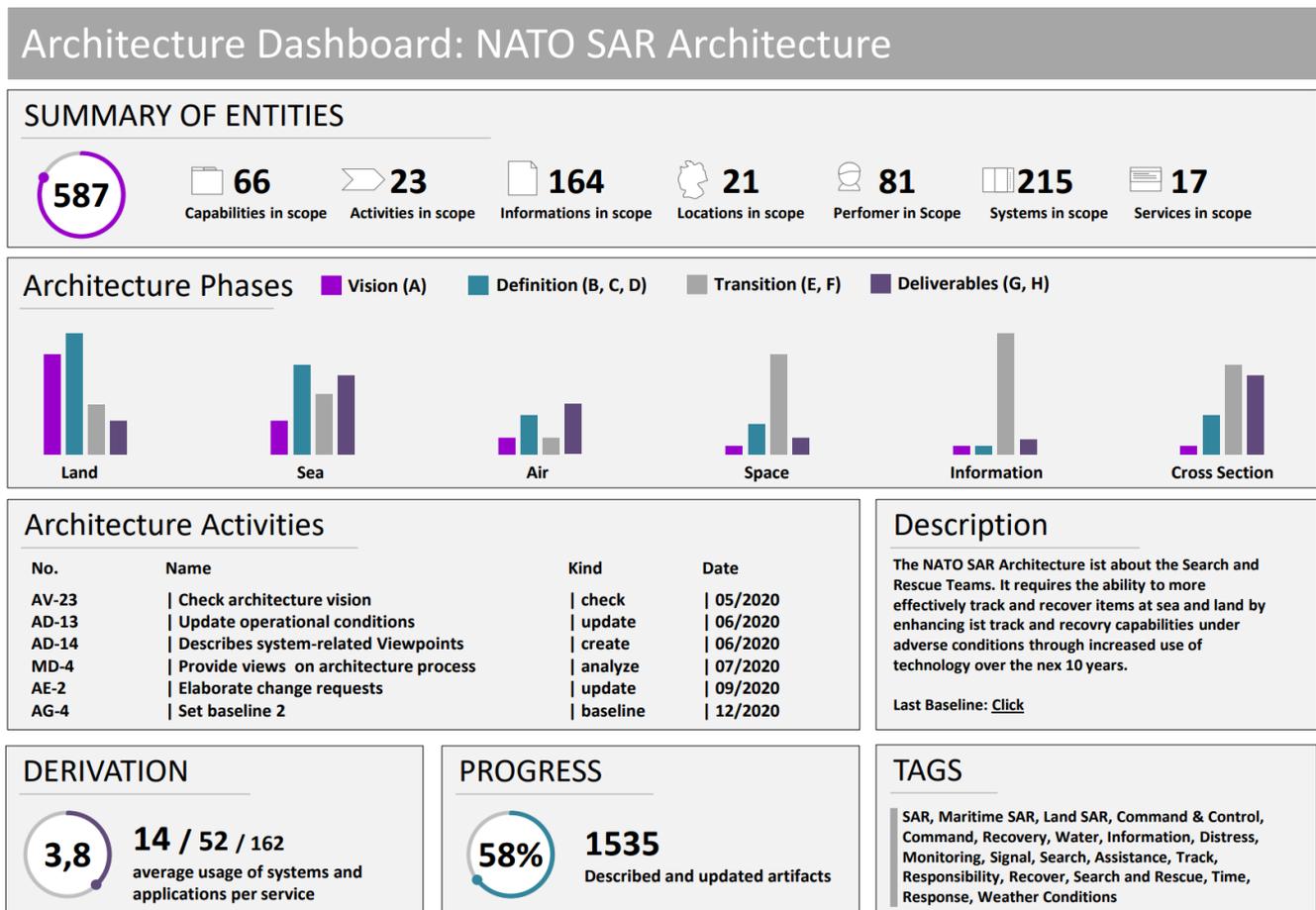
2.26.4 The body of the main architecture document describes retained architecture alternatives from stakeholder's viewpoints, and for each candidate, the set of assumptions and results interpreted to support decision-making.

2.26.5 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.

2.27 Architecture Dashboard

- 2.27.1 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.
- 2.27.2 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.
- 2.27.3 Architecture life cycle is different from projects milestones. However, projects plans include synchronization points to align with architecture evolution.
- 2.27.4 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.



2. 11 - Dashboard Example Depicting Architecture Activities and Status

- 2.27.5 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.

- 2.27.6 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).
- 2.27.7 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.
- 2.27.8 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.
- 2.27.9 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.

3 ARCHITECTING ACTIVITY

3.1 Architecting Stages

3.1.1 Table 2. 2 - Architecting Stages describes architecting activities in an architecting organization. They are organized in 8 stages, as follows:

Table 2. 2 - Architecting Stages

	Stages	Description
1	Establish Architecture Landscape (AL)	Describes the overall context and defines the capabilities and means to develop an architecture.
2	Establish Architecture Vision (AV)	Defines the architecture vision taking into account the landscape, stakes and time to market (or time to Customer).
3	Describe Alternatives of Architectures (AD)	Describes architecture from stakeholders' viewpoints according to landscape, and identify a set of alternatives of architectures for evaluation.
4	Evaluation Alternatives of Architectures and Propose Trade-Off (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	Develop and Maintain Migration Plan (MP)	Updates architecture migration plan and provides rationale for application.
6	Govern Application of Architectures (AG)	Checks the application the best architecture trade- off according to the migration plan and provide guidance to resolve dependency conflicts.
7	Decide on Architecture Changes (AC)	Elaborate and get approval on requests for architecture change.
8	Manage Architecture Motivation Data and 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).

3.1.2 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, 42020 and 42030),
- ease its deployment within various contexts, not only information technology, and
- allow flexibility in the navigation through architecting stages.



2. 12 - Architecting Stages

3.2 Method:

- 3.2.1 allows the use of any number of Viewpoint(s) and Views per architecting stage,
- 3.2.2 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,
- 3.2.3 allows more emphasis on the decision to change architecture and re-orientate the architecture due to a major evolution of motivation data, and
- 3.2.4 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.
- 3.2.5 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.
- 3.2.6 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.
- 3.2.7 The DoDAF¹⁸ architecture process, described in 2. 13, can be mapped to following stages of the

¹⁸ DODAF – USA Department of Defense Architecture Framework

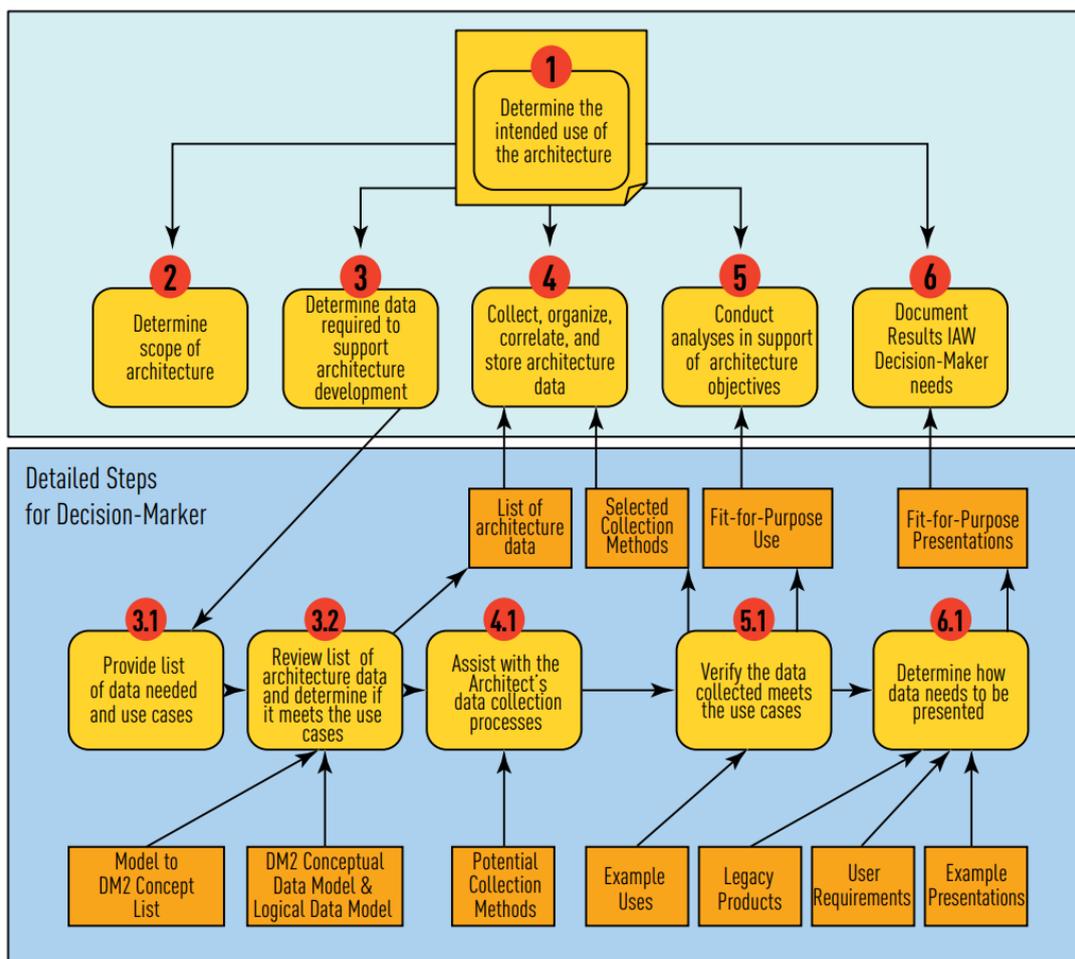
NAFv4 methodology:

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

3.2.8 The NAFv4 methodology defines eight stages see Table 2. 2, 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.

3.2.9 The method is compliant with the Six-step process for architecting introduced by DoDAF (2. 13). 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).

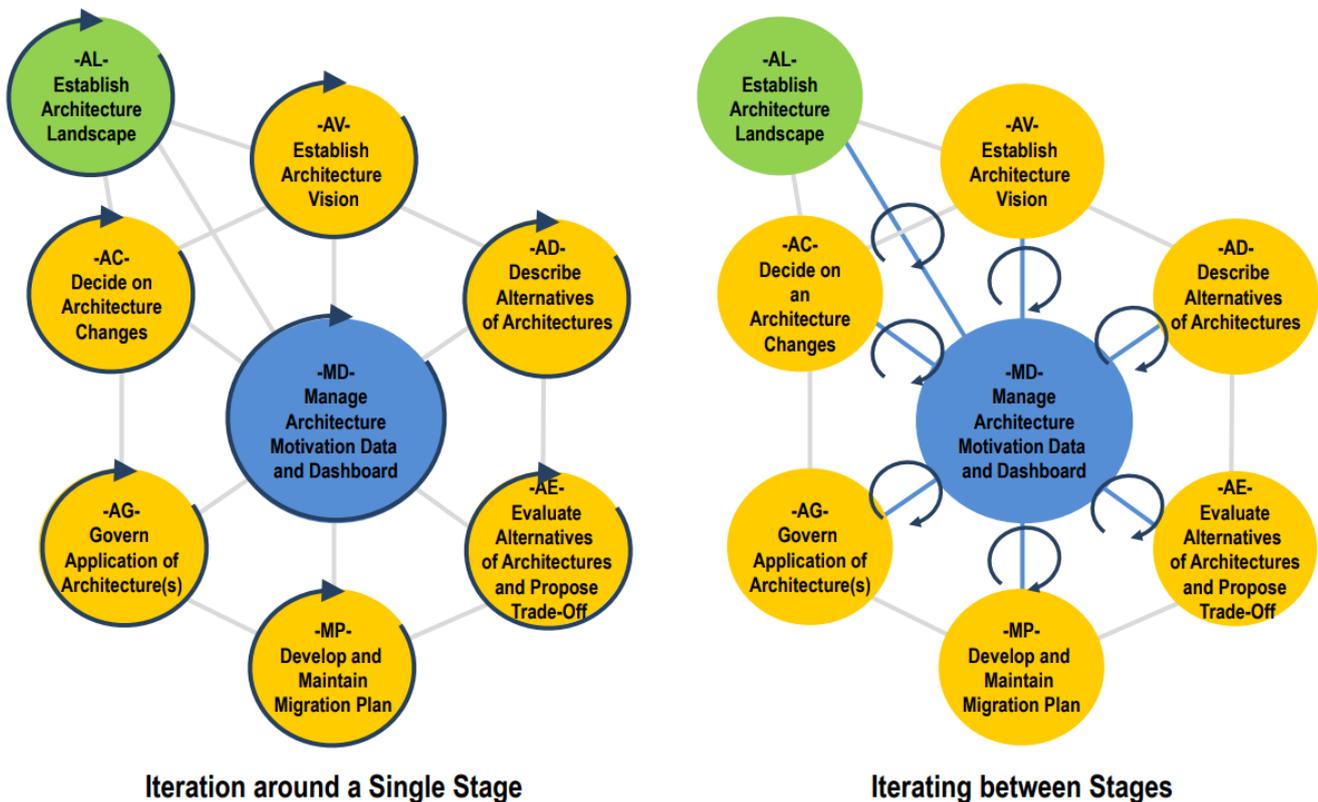


2. 13 - Six-Steps Architecture Process DoDAF v2.0

3.3 Architecting Dynamics

3.3.1 Along architecture life cycle list, 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.

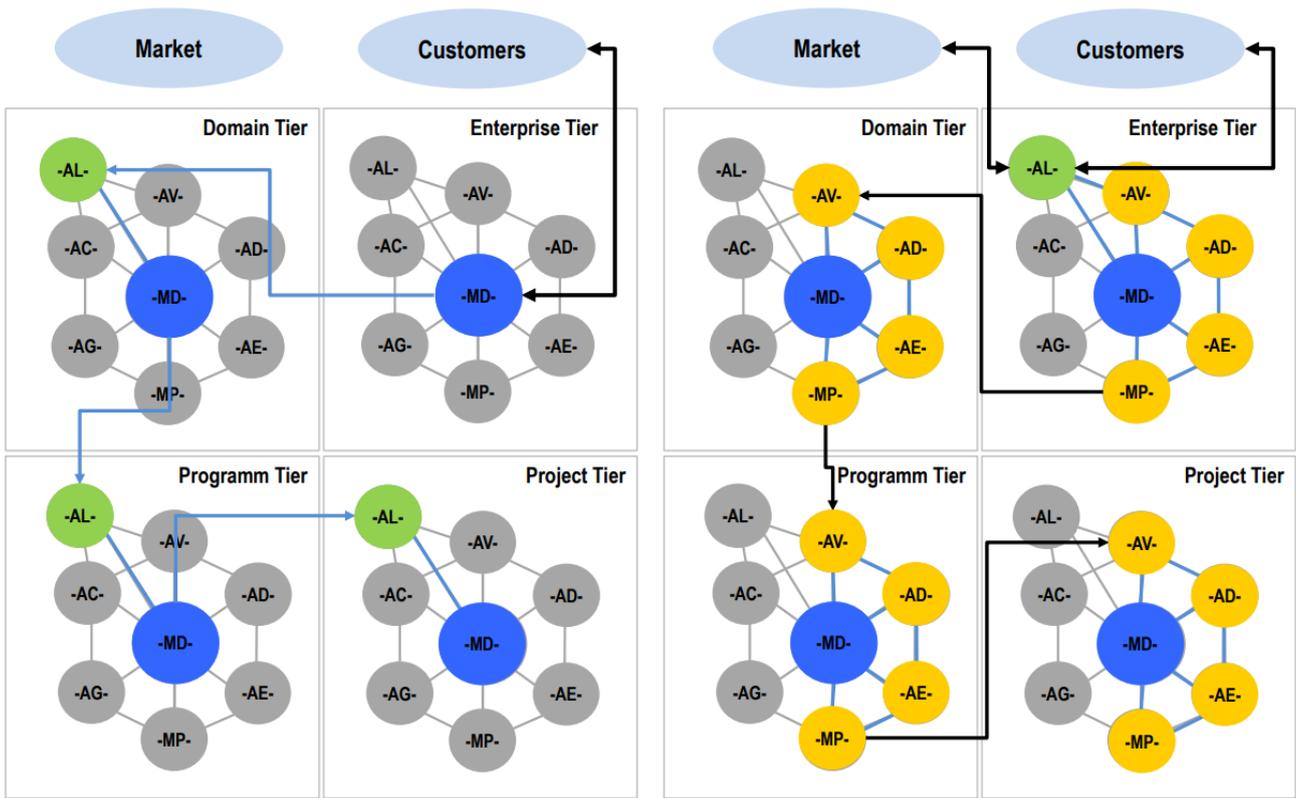
- 3.3.2 Objectives and plan of each phase are key inputs to the dashboard. Architects plan stages and define success criteria collected in the architecture motivation data. The architecture management plan captures justified cycles, iterations and synchronizations with other levels architectures.
- 3.3.3 Additional information if any (criticality, priority, weighting) on success criteria are usually submitted for approval of the governance board along trade-offs activities.
- 3.3.4 2. 14 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 2. 14 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.
- 3.3.5 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.
- 3.3.6 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.



2. 14 - Architecting Cycles & Iterations

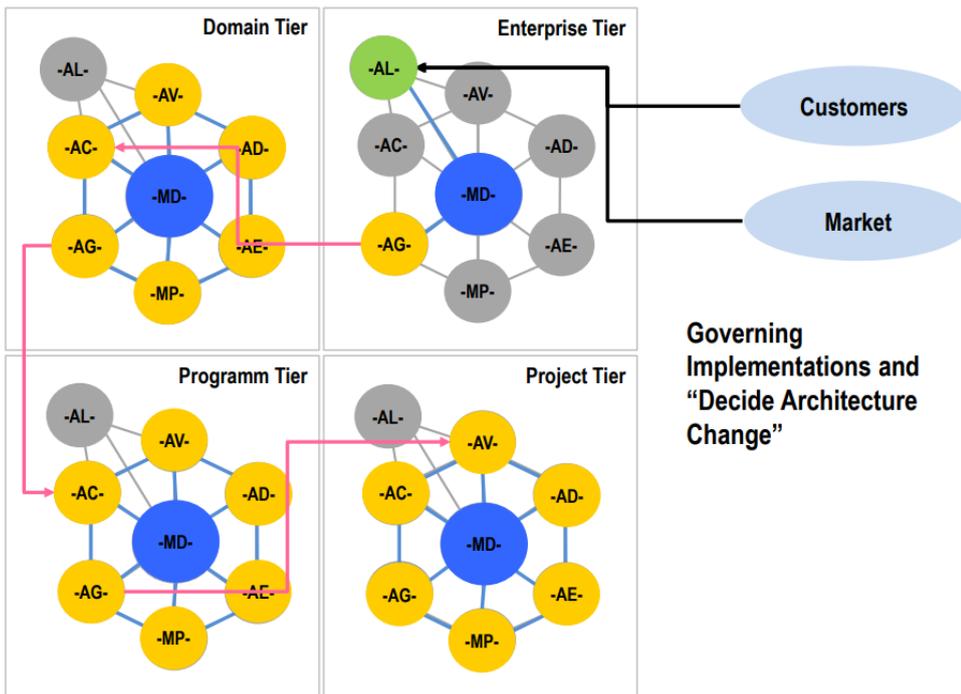
3.4 Multi-level Architecting

- 3.4.1 Architecture activities can be run by different levels: the enterprise, domains within the enterprise, and programs in enterprise domains, projects, belonging to or shared by programs or portfolios.
- 3.4.2 The architecture environment has to consider therefore target markets, customers and shareholders policies, as depicted in 2. 15. Architecture changes driven by markets and or customers trigger vision updates at enterprise level, whilst transformation will be managed and checked at different domains, starting from updates to their vision. Program and project visions are impacted accordingly.
- 3.4.3 Landscapes are updated from enterprise down to projects, and from projects up to enterprise, to enable overall governance of enterprise transformation.



Motivation Data Change

Impact of Migration Planning on Current Local Vision



Governing Implementations and "Decide Architecture Change"

2.15 - Architecting Environment

4 ARCHITECTING FOR THE ENTERPRISE SCOPE

4.1 Introduction

- 4.1.1 Architecture elaborated to master the overall enterprise business are typically:
- 4.1.2 Architecture of the enterprise itself. The enterprise is therefore analyzed 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.
- 4.1.3 Architectures used by the programs and the projects of the enterprise in order to deliver the enterprise systems/products required by internal and external contracts.
- 4.1.4 In both cases these architectures provide directions and guidance for the enterprise programs and projects in charge of developing and maintaining either the enterprise itself or the enterprise systems/products.
- 4.1.5 These architectures are considered an input for enterprise governance.

4.2 Overview of the Enterprise Architecting Stages

Table 2.3 - Overview of the Enterprise Architecting Stages

	Stages	Description
1	Enterprise: Establish 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: Establish 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: Describe Alternatives of Architectures (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: Evaluate Alternatives of Architectures and Propose Trade-Off (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: Develop Migration Plan (MP)	Get an updated transformation roadmap for application of the Enterprise Architecture with a rationale and a governance model.
6	Enterprise: Govern Application of Architectures (AG)	Check for the application of the Enterprise Architecture according to the migration plan and provide recommendation.
7	Enterprise: Decide on 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: Manage Architecture Motivation Data and 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.



The content of the Enterprise Architecting Stages is also applicable to architectures at the Capability Level.

4.3 Enterprise Architecting Activities

Table 2. 4 - Enterprise: Establish Architecture Landscape (AL)

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

Table 2. 5 - Enterprise: Establish Architecture Vision (AV)

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

Table 2. 6 - Enterprise Describe Alternatives of Architectures (AD)

Enterprise: Describe Alternatives of Architectures (AD)	
Objectives	Task
<p>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.</p>	<p>Analyze 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.</p>
Inputs	Outputs
<p>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.</p>	<p>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).</p>
Recommended Views	Stakeholders
<p>C1 to 8, Cr, S1 to 8, Sr, L1 to 8,Lr, P1 to 8, Pr, A1, A2, L2-L3 (ACD), A8.</p>	<p>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.</p>

Table 2.7 - Enterprise: Evaluate Alternatives of Architectures and Propose Trade-Off (AE)

Enterprise: Evaluate Alternatives of Architectures and Propose Trade-Off (AE)	
Objectives	Task
<p>To formalize the evaluation criteria according to the concerns of the stakeholders,</p> <p>To evaluate each candidate Enterprise Architecture,</p> <p>To evaluate the risk, cost, value and opportunities for each Enterprise Architecture,</p> <p>To select the Enterprise Architectures for application.</p>	<p>Define the evaluation objectives from the Enterprise Architecture vision,</p> <p>Refine the list of stakeholders, their concerns and questions with regards to the enterprise motivation data,</p> <p>Define the evaluation criteria from the concerns of the stakeholders, with their relative importance (priorities, weights, etc.),</p> <p>Determine techniques, methods and tools for performing the evaluation,</p> <p>Evaluate each architecture alternative with collection and understanding of required information (metrics),</p> <p>Formulate the findings per architecture alternative,</p> <p>Perform trade-off analysis with estimate of risk, cost, value and opportunities,</p> <p>Choose the best alternatives of Enterprise Architectures with rationale against the enterprise motivation data,</p> <p>Perform gap analysis between the evaluation objectives and the achieved architecture evaluation,</p> <p>Finalize and review the Enterprise Architecture evaluation results.</p> <p>Request for change of the alternatives of architectures as necessary,</p> <p>Create architecture evaluation document for this iteration.</p>
Inputs	Outputs
<p>Request for architecture work with a statement of work,</p> <p>Enterprise Architecture vision (list of stakeholders, concerns and questions),</p> <p>Enterprise motivation data,</p> <p>Architecture principles,</p> <p>Pre-existing Enterprise Architecture, evaluation elements in the Enterprise Architecture repositories,</p> <p>Enterprise Architecture landscape,</p> <p>Enterprise Architectures descriptions.</p>	<p>Reviewed selection of Enterprise Architectures with assessment of risk, cost, value and opportunities,</p> <p>Architecture evaluation document including objectives, criteria, evaluation results and selection,</p> <p>Gaps with regards to Enterprise Architecture vision (gaps with the evaluation objectives),</p> <p>Requests for changes of the alternatives of architectures.</p>
Recommended Views	Stakeholders

<p>C1 to C8, Cr, S1 to 8, Sr, L1 to 8,Lr, P1 to 8, Pr, A1, A2, L2-L3 (ACD), A8.</p>	<p>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.</p>
---	---

Table 2. 8 - Enterprise: Develop and Maintain Migration Plan (MP)

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

Table 2. 9 - Enterprise: Govern Application of Architectures (AG)

Enterprise: Govern Application of Architectures (AG)	
Objectives	Task
<p>To ensure correct application of the Enterprise Architectures in the enterprise transformation, To provide recommendation towards the governance authority of the enterprise transformation.</p>	<p>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.</p>
Inputs	Outputs
<p>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 program.</p>	<p>Governance model (directive and guidance) for application of the Enterprise Architectures, Corrective recommendation for applications of the Enterprise Architectures, Change requests for Enterprise Architectures.</p>
Recommended Views	Stakeholders
<p>A1 to A8, Ar, C1, C2, S1, S2, L1, L2, L2-L3, P1, P2.</p>	<p>The governance model is proposed by the Board of Enterprise Architects, Outputs are agreed by the Enterprise Architecture Governance Board.</p>



For governance activities, it is highly recommended to consider COBIT¹⁹ and ISO 38500²⁰ in addition to NAF Chapter 2.

¹⁹COBIT (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).

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

Table 2. 10 - Enterprise: Decide on Architecture Changes (AC)

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

Table 2. 11 - Enterprise: Manage Architecture Motivation Data and Dashboard (MD)

Enterprise: Manage Architecture Motivation Data and Dashboard (MD)	
Objectives	Task
<p>To manage a consistent access to the enterprise motivation data.</p> <p>To provide consistent architecture dashboard related to activities, Enterprise Architecture landscape (including Enterprise Architectures in repositories) and enterprise resources.</p>	<p>Manage updates of the enterprise motivation data asked by the Enterprise Architecture Governance Board and those coming from the Enterprise Architecture stages,</p> <p>Analyze enterprise external and internal architectures and architecture elements able to enrich the enterprise Architecture repositories.</p> <p>Update the repositories, as necessary,</p> <p>Analyze enterprise external and internal references able to enrich enterprise reference libraries. Update the libraries, as necessary,</p> <p>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,</p> <p>Manage a consistent access to enterprise motivation data,</p> <p>Report to the Enterprise Architecture Governance Board.</p>
Inputs	Outputs
<p>Enterprise request for update of the enterprise motivation data,</p> <p>Enterprise external and internal architectures and architecture elements,</p> <p>Enterprise external and internal references,</p> <p>Organizational model for Enterprise Architecture,</p> <p>Enterprise Architecture landscape.</p>	<p>Request for update of Enterprise Architecture landscape,</p> <p>Updated enterprise motivation data,</p> <p>Report to the Enterprise Architecture Governance Board.</p>
Recommended Views	Stakeholders
<p>A1, A7, A5.</p>	<p>The enterprise motivation data are proposed by the Board of Enterprise Architects,</p> <p>Outputs are agreed by the Enterprise Architecture Governance Board.</p>

5 ARCHITECTING IN A PROJECT

5.1 Overview of Project Architecting Activities

- 5.1.1 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.
- 5.1.2 Evaluation criteria are initialized from vision elements, namely architecture objectives.
- 5.1.3 The description stage identifies and describes alternatives of architectures which satisfy key architecture requirements and known constraints.
- 5.1.4 The evaluation stage provides support to decision-making, using criteria agreed by stakeholders.

5.2 Project Architecting Activities

Table 2. 12 - Project: Establish Architecture Landscape (AL)

Project: Establish Architecture Landscape (AL)	
Objectives	Task
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.	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, analyze 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.
Inputs	Outputs

<p>Context, drivers and constraints calling for architecture capability:</p> <ul style="list-style-type: none"> - 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. 	<p>Organization of architecture team: architecture OBS and agreed workflow from vision to architecture baseline,</p> <p>Tailoring of the architecting process to enable the workflow,</p> <p>Definition of resources, skills and roles according to the tailored architecting process,</p> <p>Definition of key interfaces to complementary architecture frameworks if any (i.e., dedicated architecture framework),</p> <p>Definition and statement of work for customization & initialization of architecture support tools, including interfaces to complementary tools and repositories if any,</p> <p>Agreement on architecture principles applicable from vision to baseline. Agreed principles (or links to) are initialized within the architecture repositories at kick-off,</p> <p>Architecture management plan outline.</p>	
Stakeholders	Input Views	Output Views
<p>Architect, Project Manager, Representatives of plans, operations, legacy systems, standards, technology, regulations and laws), Specialists (security, safety, human factors, etc.), Sponsors.</p>	<p>L2, L3, L4, P3, P4, P8, Pr, A8.</p>	<p>A1 to 8 (i.e. foundation, architecture plan and architecture summary documentation with references to input views.</p>

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

Project: Establish Architecture Vision (AV)		
Objectives	Task	
<p>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.</p>	<p>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, Analyze 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.</p>	
Inputs	Outputs	
<p>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.</p>	<p>(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.</p>	
Stakeholders	Input Views	Output Views
<p>Architect, Project Manager, Representatives of plans, operations, legacy systems, standards, and technology.</p>	<p>A1 to A8, Ar. C1 to C8, Cr. S1, Sr, L1, Lr, P1, Pr (of legacy).</p>	<p>A1 to 8 (i.e. foundation, architecture plan and architecture summary documentation with references to input views.</p>

Table 2. 14 - Project: Describe Alternatives of Architectures (AD)

Project: Describe Alternatives of Architectures (AD)		
Objectives	Task	
Describe, starting from ‘as-is’ architecture and, in consistency with Enterprise Architecture principles, alternatives of solution architectures that meet project’s architecture vision.	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.	
Inputs	Outputs	
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, TEPIDOIL21, 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.	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.	
Stakeholders	Input Views	Output Views
Architect, Project Manager, Representative of: plans, operations, legacy systems, standards, technology regulations and laws, Security architect, safety architect.	A1 to A8, Ar. C8, Cr. S1, Sr, L1, Lr, P1, Pr (of legacy).	A1 to A8, Ar. C1 to C8, Cr.

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

Table 2. 15 - Project: Evaluate Alternatives of Architectures and Propose Trade-Off (AE)

Project: Evaluate Alternatives of Architectures and Propose Trade-Off (AE)		
Objectives	Task	
<p>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.</p>	<p>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.</p>	
Inputs	Outputs	
<p>Statement of architecture work. Initialized evaluation and comparison grid, Weighted comparison criteria</p>	<p>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.</p>	
Stakeholders	Input Views	Output Views

<p>Architect, Project Manager, Representative of: plans, operations, legacy systems, standards, technology, regulations and laws, Security architect and safety architects, Representative of human factors Sponsor.</p>	<p>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.</p>	<p>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).</p>
--	---	---

Table 2. 16 - Project: Develop and Maintain Migration Plan (MP)

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

Table 2. 17 - Project: Govern Application of Architectures (AG)

Project: Govern Application of Architectures (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 2. 18 - Project: Decide on Architecture Changes (AC)

Project: Decide on Architecture Changes (AC)		
Objectives	Task	
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.	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.	
Inputs	Outputs	
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.	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.	
Stakeholders	Input Views	Output Views
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.	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. A1-8 and/or C1-8 and S1-8 or and /or L1-8, P1-8, Ar, Cr, Sr, Lr, Pr.

Table 2. 19 - Project: Manage Architecture Motivation Data and Dashboard (MD)

Project: Manage Architecture Motivation Data and Dashboard (MD)	
Purpose	Tasks
<p>Set and maintain architecture up-to-date motivation data, Monitor architecture progress and stop activities according to enterprise policy and stakeholder's expectations.</p>	<p>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, DOTMLPFI, 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 NAF views, 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.</p>
Inputs	Outputs

<p>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.</p>	<p>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.</p>	
Stakeholders	Input views	Output views
<p>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.</p>	<p>C1 to C3, Cr, Lr, A1 to A8.</p>	<p>Updates of C1 to C3, A1 to A8, Ar, Cr, Sr, Lr, Pr.</p>

6 FOUNDATION FOR ARCHITECTING

6.1.1 This section describes the common methodological elements necessary to elaborate either Enterprise or Project Architecture Frameworks.

6.1.2 These elements are related to activities and architecture data:

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

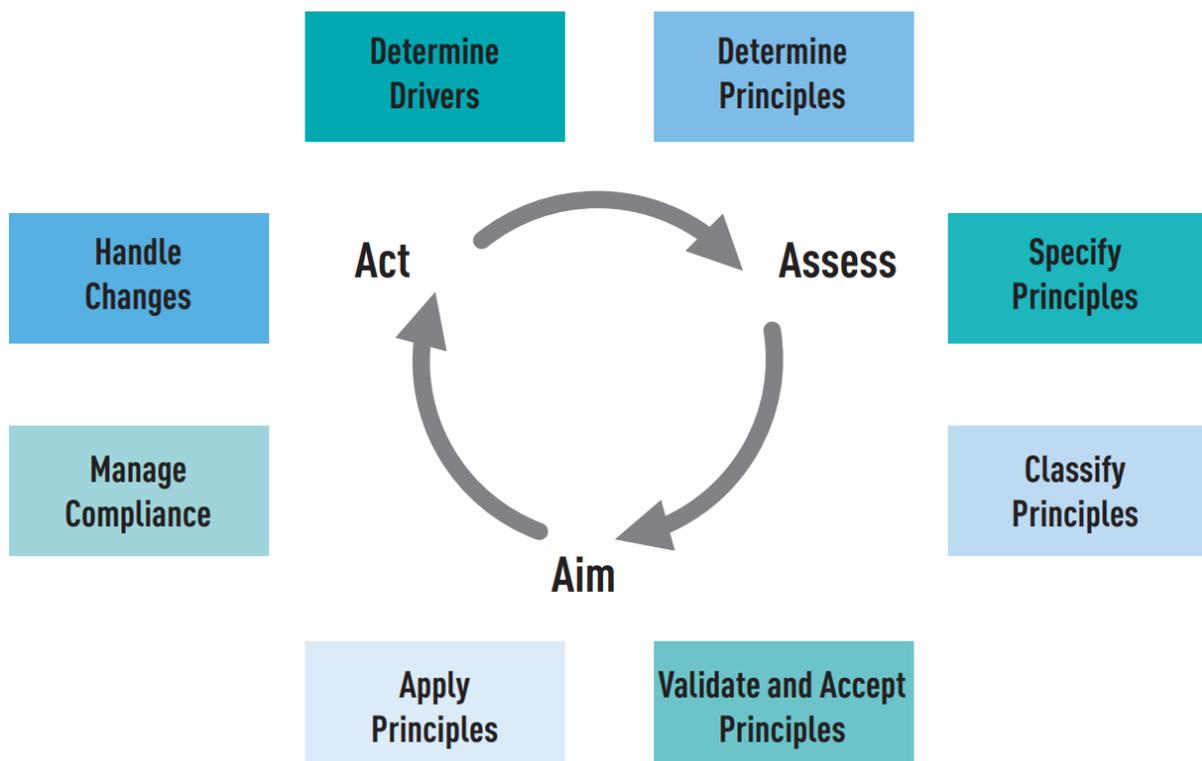
6.2 Architecting Principles (Foundation for Best Practices)

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

6.3 Overview

6.3.1 2.16 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.



2. 16 Architecture Principles Definition and Management Activities

Table 2. 20 - Principle Kinds

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.

6.4 Description of Sub-Processes

6.4.1 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. 21 - Level of Compliance

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

- 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.

6.5 Architecture Principles in NAFv4

- 6.5.1 The Architecture activities for both enterprise and projects are grouped in 8 stages. These stages are all concerned with architecture principles.
- 6.5.2 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).
- 6.5.3 Changes to them are handled during the stage dealing with the decisions on the architecture change (AC).
- 6.5.4 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,
- 6.5.5 These principles are derived and adapted for the architecture activities in the projects according to the architecture motivation data in these projects,

- 6.5.6 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,
- 6.5.7 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,
- 6.5.8 Architecture principles should have a name, statement, rationale and implications,
- 6.5.9 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,
- 6.5.10 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,
- 6.5.11 Also, it may work upon architecture principles that are specific to the architecture perspectives like: business architecture principles and data architecture principles.

6.6 Architecture Capabilities

- 6.6.1 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),
- 6.6.2 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
- 6.6.3 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.

6.7 Recommended Patterns for Architecture and Architecting

- 6.7.1 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,
- 6.7.2 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
- 6.7.3 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.

6.8 Architecture Assets

- 6.8.1 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,
- 6.8.2 The architecture assets basically include deliverable and building blocks. Architecture patterns can also be considered as assets to some extent. However assets cannot be 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.
- 6.8.3 A real ontology is needed here to describe formally the Architecture Data,
- 6.8.4 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.
- 6.8.5 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 are 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/IEEE 42010 provides the following definitions relevant to this chapter:

Table 3. 22 - Relevant Descriptions

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 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 NAF viewpoints, 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 required.

	Taxonomy		Structure		Behaviour				Information	Constraints	Roadmap
	Connectivity	Processes	States	Sequences							
Concepts	C1 Capability Taxonomy	C2 Enterprise Vision	C3 Capability Dependencies	C4 Standard Processes	C5 Effects			C7 Performance Criteria	C8 Planning Constraints	Cr Capability Roadmap	
	C1-S1										
Service Specifications	S1 Service Taxonomy	S2 Service Structure	S3 Service Interfaces	S4 Service Functions	S5 Service States	S6 Service Sequence	S7 Service Interface Parameters	S8 Service Constraints	Sr Service Roadmap		
Logical Specifications	L1 Logical Taxonomy	L2 Logical Structure	L3 Logical Interactions	L4 Logical Activities	L5 Logical States	L6 Logical Sequence	L7 Information Model	L8 Logical Constraints	Lr Logical Roadmap		
			L4-P4								
Physical Resource Specifications	P1 Resource Taxonomy	P2 Resource Structure	P3 Resource Interactions	P4 Resources Functions	P5 Resources States	P6 Resource Sequence	P7 Data Model	P8 Resource Constraints	Pr Resource Roadmap		
Architecture Foundation	A1 Metadata Definitions	A2 Architecture Products	A3 Architecture Correspondance	A4 Architecture Methodology	A5 Architecture Status	A6 Architecture Versions	A7 Architecture Metadata	A8 Architecture Standards	Ar Architecture Roadmap		

3. 17 - NAF Grid Representation

2.1.2 The grid approach presents the NAF viewpoints 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 2.

Table 3. 23 - Descriptions of the Columns in the Grid

Aspects	Description
Taxonomy	Specialization hierarchies of architecture elements such as capabilities, services, etc.
Structure	How structural elements are assembled (enterprises, nodes, resources, etc.).
Connectivity	Everything from high-level capability dependencies to detailed system connectivity.
Behaviour	How things work: Processes - Procedures flows and their decomposition and responsibilities. States – Staes and Modes of structural or behavioural elements. Sequences - How things interact and in what order.
Information	Addresses relevant KPIs and Information.
Constraints	Rules that govern the enterprise, nodes, resources, etc.
Roadmap	Temporal information about the entity of interest or its architecture.

- 2.1.3 The NAF viewpoints retain an equivalence with the NAFv3 Views²², albeit with names that better describe their purpose, as indicated in Table 3-3.

Table 3. 24 - 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 Foundation (A)

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

2.2 NAF Information Model

- 2.2.1 The NAF Information Model (NAF IM) defines the conceptual foundation of NAFv4 by describing the architectural concepts and relationships that ensure consistency across all viewpoints.
- 2.2.2 It is not intended for direct implementation. Instead, the NAF IM provides the conceptual basis for the UAF DMM and ArchiMate Implementation Guides for NAFv4, which define how the framework is realized within those respective modelling approaches. By separating conceptual definition from implementation, the NAF IM provides a coherent and unified foundation for all NAF-compliant modelling practices.

2.3 Approach

- 2.3.1 Each viewpoint within the NAF is presented, in its own subsection, using a consistent structure to ensure clarity, comparability, and traceability across the framework. This section explains the format and content of each viewpoint and how it should be interpreted.
- 2.3.2 Each subsection includes a Cut-out from the NAF Information Model, showing the semantics relevant to that viewpoint.
- 2.3.3 All viewpoint examples in this document are based on a typical capability of a military organization and a common operational scenario - Search and Rescue (SAR). The SAR scenario provides a realistic, multi-domain context that involves coordination between civil and military actors, supporting systems and services, and communication networks. It offers a relatable use case for illustrating how the NAF Information Model can be applied across the different architectural domains in a consistent way.

²²

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

2.3.4 Structure of the NAF Viewpoint documentation:

- **Purpose** - Outlines the intent of the viewpoint, the rationale for its inclusion within the framework and the types of architectural insight it is designed to convey. It summarizes why it is relevant to stakeholders and distinguishes it from other viewpoints.
- **Usage and Concerns Addressed** - Identifies the typical use cases the viewpoint is supporting and specifies the stakeholder concerns it is intended to address, in alignment with the ISO/IEC/IEEE 42010 definition of an architecture viewpoint. It may also describe when in the architecture lifecycle the viewpoint is most applicable (e.g., capability planning, system design, or implementation).
- **Scope** - Defines the boundaries of what the viewpoint depicts. It lists the mandatory and optional NAF IM concepts and relationships to ensure the viewpoint remains focused and unambiguous. It is derived from the combination of the **Purpose** and **Usage and Concerns Addressed**, ensuring that each viewpoint includes only the concepts necessary to address its intended concerns. Optional elements on a given viewpoint are there to provide more context to the viewpoint using objects from other columns or to show traceability with viewpoints from other rows, or columns, of the grid.
- **Representation** - Describes how the viewpoint should be represented when used to communicate with stakeholders and address their specific concerns. The representation focuses on clarity and readability, presenting information in a way that is meaningful and accessible to its intended audience rather than as a detailed diagram. While the focus of the NAF IM is on how to create formalized models of the information relevant to the entity of interest, the representation makes use of the data in a way the consumer is able to understand it seamlessly.

2.3.5 Viewpoint Cut-out:

- **Cut-outs** - Provide a visual excerpt of the NAF IM corresponding to the scope of the viewpoint. The Cut-out shows the concepts and relationships defined as mandatory or optional within that scope. It may include abstract elements for simplicity rather than displaying all possible relationships to sub-types (for example, the Logical Active Resource on the L1 Viewpoint).
- **Cut-out Notes** - Where applicable, contains explanatory notes supporting the viewpoint Cut-out, clarifying modelling rules, element definitions, and any relevant dependencies or assumptions. These notes assist the implementation of metamodels and modelers in applying the viewpoint correctly.

2.3.6 Viewpoint Example:

- **NAFv4 Compliant Example** - For each viewpoint an example based on the Search and Rescue (SAR) scenario introduced above is provided. The examples demonstrate how the mandatory concepts and relationships defined for each viewpoint must be modelled - using the NAF IM. This provides a practical reference of how viewpoints can be expressed in compliant implementation modelling guides. These examples are illustrative rather than exhaustive; they are not intended to show every possible object or relationship as no single model would include all concepts of the Information Model.
- **Example Notes** - Where applicable, the example notes provide commentary on the NAFv4 compliant example help readers interpret the example correctly.

2.4 Graphical and Typographical Interpretation

2.4.1 The following styles are used within this document for viewpoint Cut-outs and examples:

- Thick borders for elements or thick lines for relationships indicate that these concepts are mandatory²³ on a specific viewpoint. This aligns to the “Shall Contain” in the scope.
- Thin borders for elements or thin lines for relationships indicate that these concepts are optional on a specific viewpoint. This aligns to the “May Show/ May Trace” in the scope.
- The arrowhead sitting on one end of a relation line indicates *reading direction* (all relations are directed).
- Element names in *italic* are abstract, meaning they cannot be directly instantiated in a model.²⁴
- Relation names are verbs or verb phases that provide the logical linking between the elements.
- Example elements have a <<stereotype>> label that identifies their corresponding IM element.

2.4.2 Within the glossary:

- Terms in brackets [...] denote a reference to another definition.
- Terms in braces {...} denote a reference to the source of the definition.

2.4.3 Each architectural domain is represented by a pair of colours: a Header Colour, taken directly from the NAF grid palette, and a corresponding Element Colour, a lighter tone used in viewpoint Cut-outs and examples. This pairing ensures that all diagrammatic content remains visually aligned with the NAF grid. Generic (abstract) elements that are not part of a specific domain are light grey. The colours used are shown below:

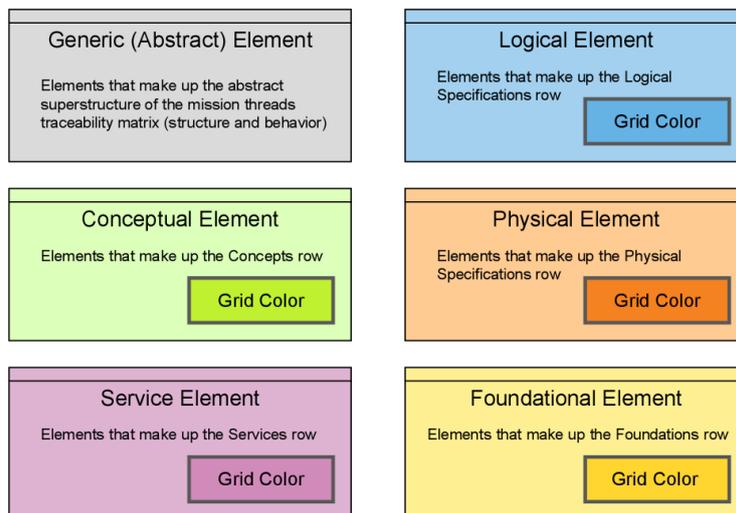


Figure 18 - Domain Colour Scheme

²³ Elements or relationships can be mandatory on one viewpoint and optional on another viewpoint.

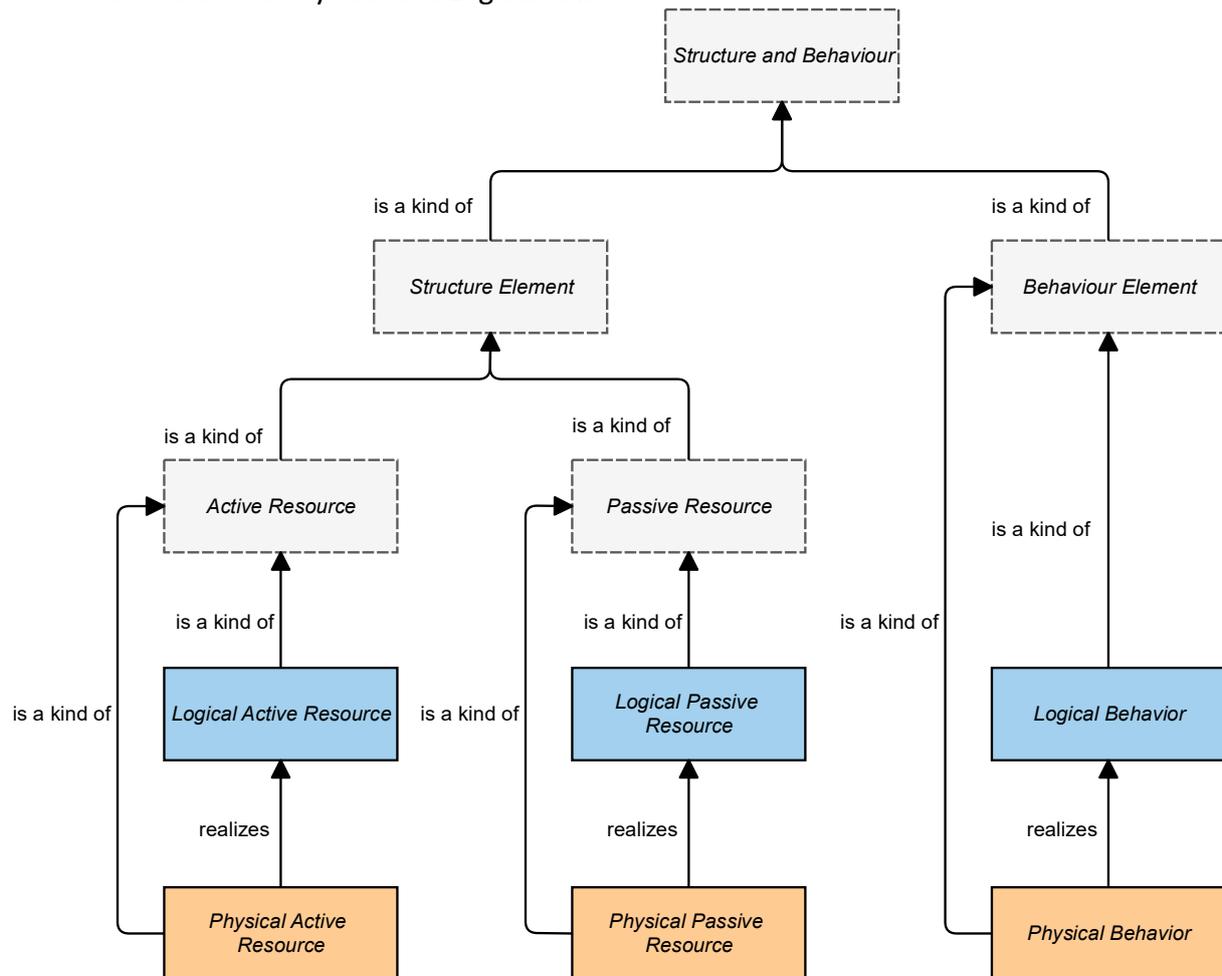
²⁴ There may be exceptions to this where an element is abstract and part of a domain which will be elaborated in relevant implementation guides.

3 SUPER STRUCTURE

3.1 Abstract Superstructure

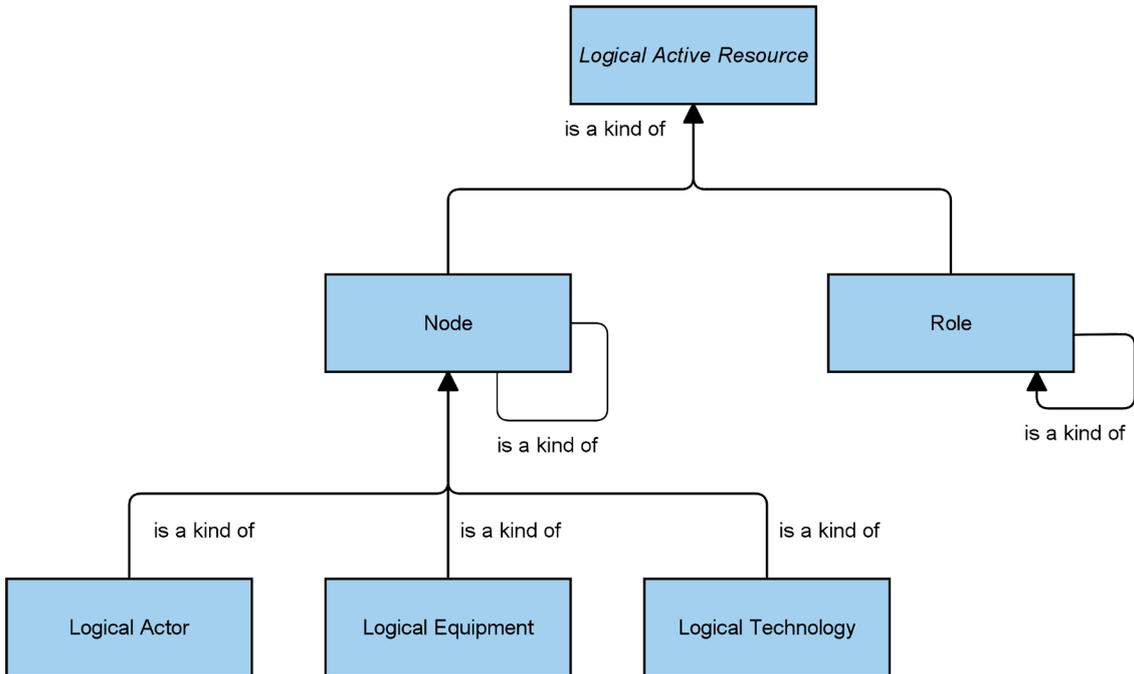
3.1.1 The NAF Superstructure provides a conceptual overview of how the main architectural domains within the NAF Information Model, and their sub-types, relate to each other.

3.1.2 The diagram below presents the core conceptual pattern at the heart of the Superstructure. It distinguishes between Structure and Behaviour, and the separation of this at the Physical and Logical rows.



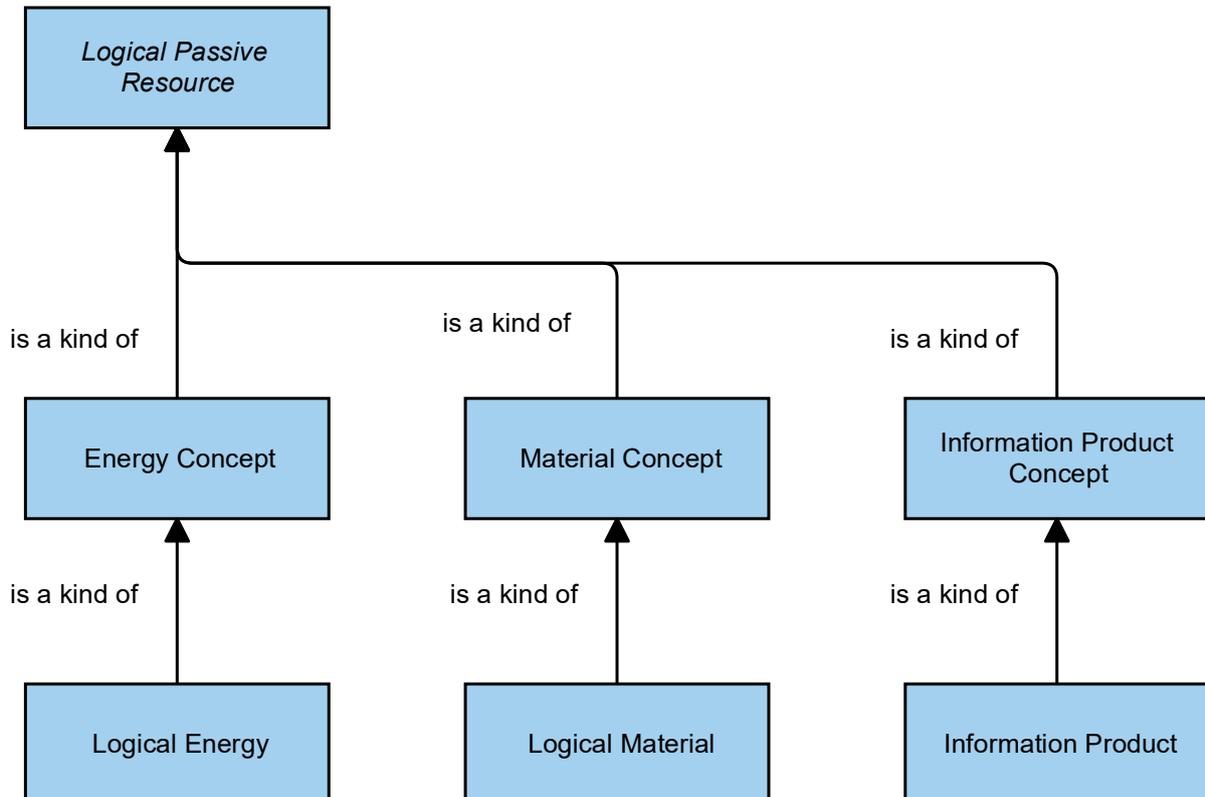
3.2 Logical Active Resources

3.2.1 The *Logical Active Resource* Supertype breaks down into:



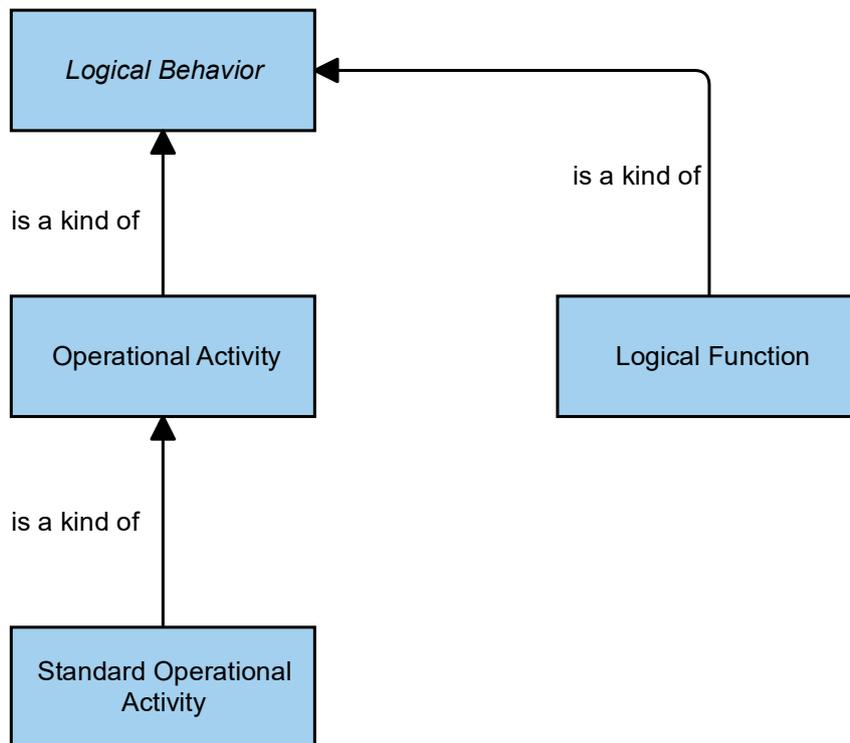
3.3 Logical Passive Resources

3.3.1 The *Logical Passive Resource* Supertype breaks down into:



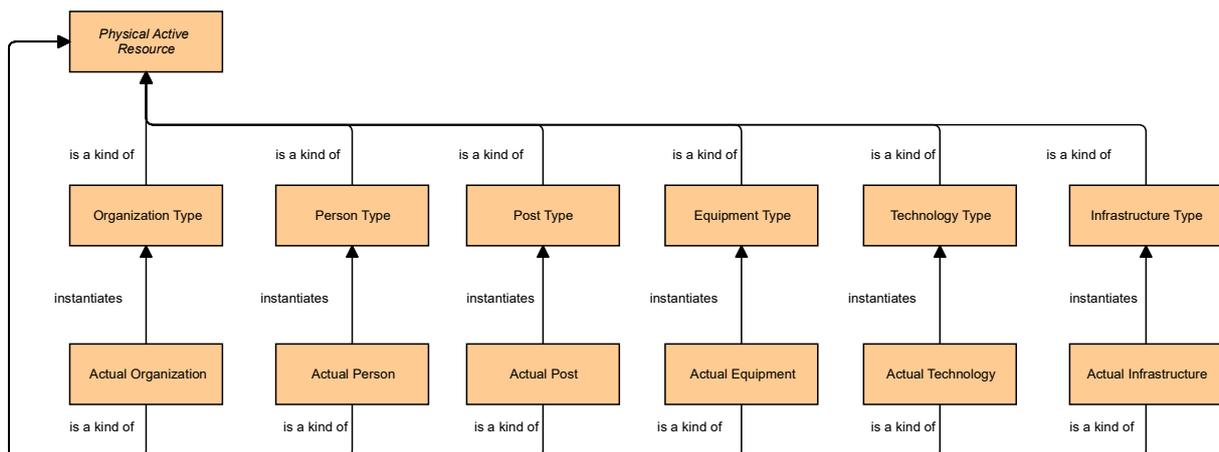
3.4 Logical Behaviour

3.4.1 The *Logical Behaviour* Supertype breaks down into:



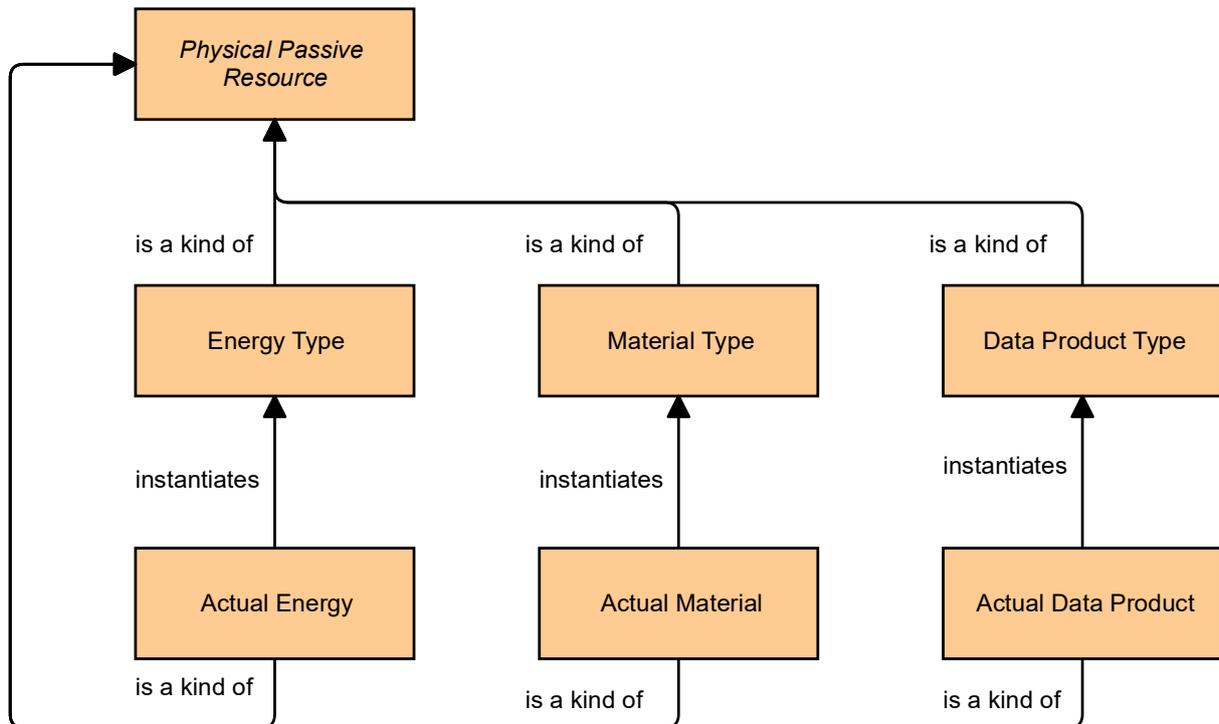
3.5 Physical Active Resources

3.5.1 The *Physical Active Resource* Supertype breaks down into:



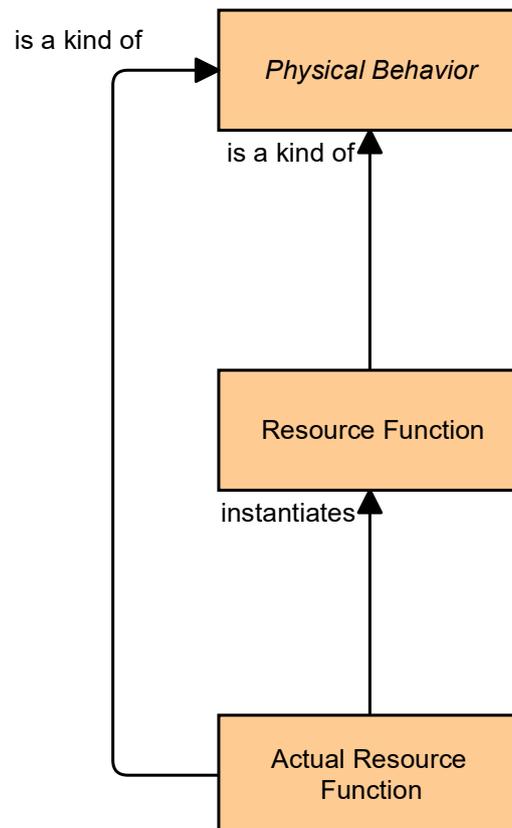
3.6 Physical Passive Resources

3.6.1 The *Physical Passive Resource* Supertype breaks down into:



3.7 Physical Behaviour

3.7.1 The *Physical Behaviour* Supertype breaks down into:



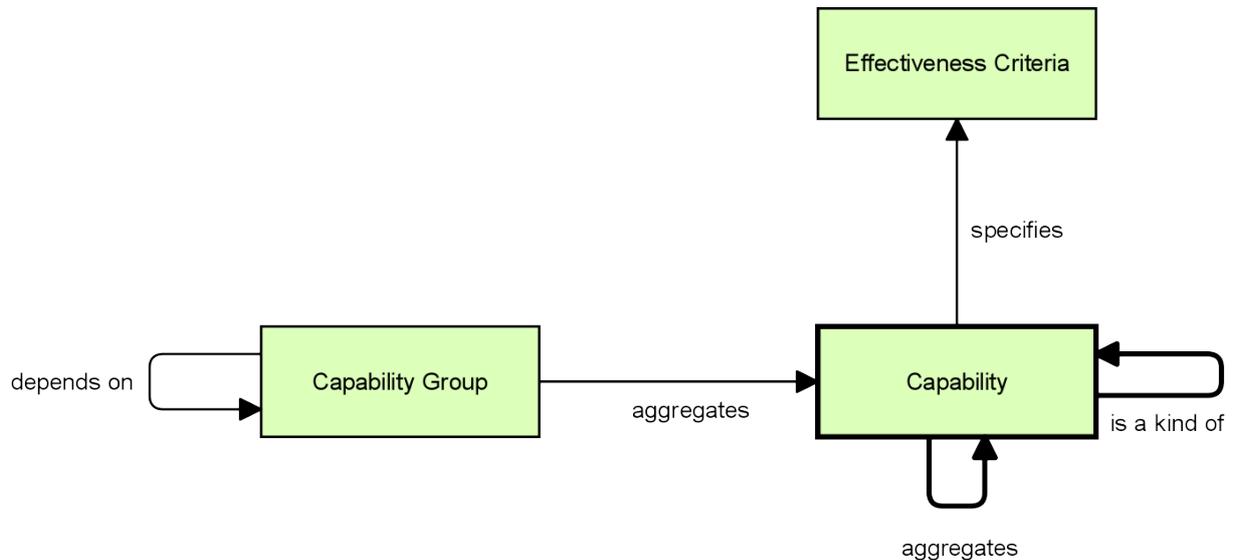
4 C1 - CAPABILITY TAXONOMY

Purpose	Usage & Concerns Addressed
The C1 Viewpoint is concerned with the identification of capabilities and their organization into specialization taxonomies, independent of their implementation.	<ul style="list-style-type: none"> • Capability Planning and Management. • Identification of existing and required Capabilities. • Providing reference Capabilities for multiple architectures.

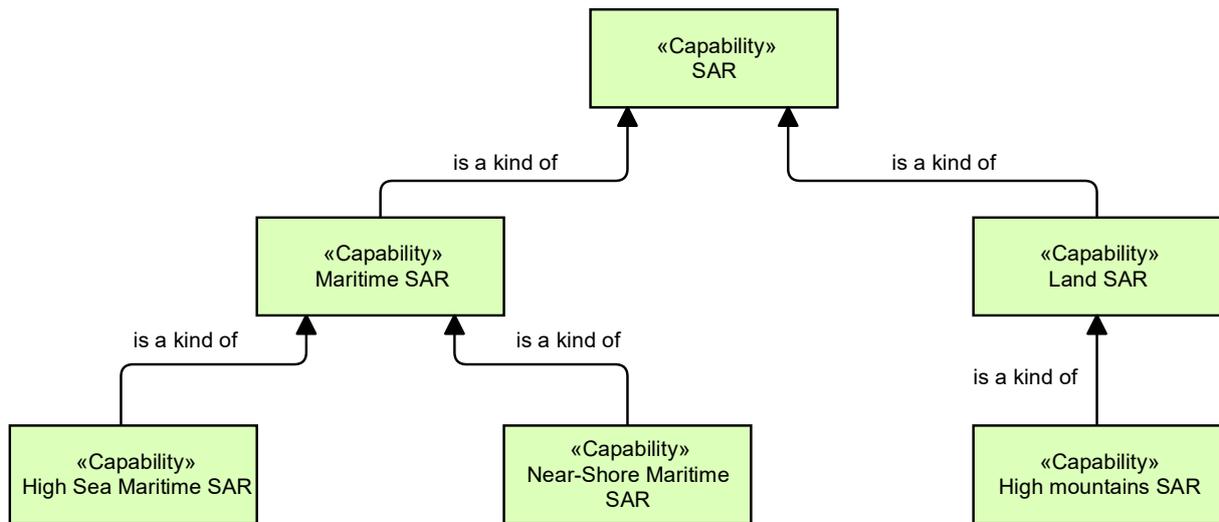
Scope
<ul style="list-style-type: none"> • Shall contain Capabilities relevant for the entity of interest • Shall contain Capabilities organized into specialization and hierarchical (aggregation) taxonomies. • May show Capabilities organized into Capability Groups. • May show Effectiveness Criteria for the Capabilities.

Representation
<ul style="list-style-type: none"> • Tabulation. • Hierarchical (Connected Shapes). • Diagram (with generalization relationships and property definitions).

4.1 Cut-out



4.2 NAFv4 Compliant Example



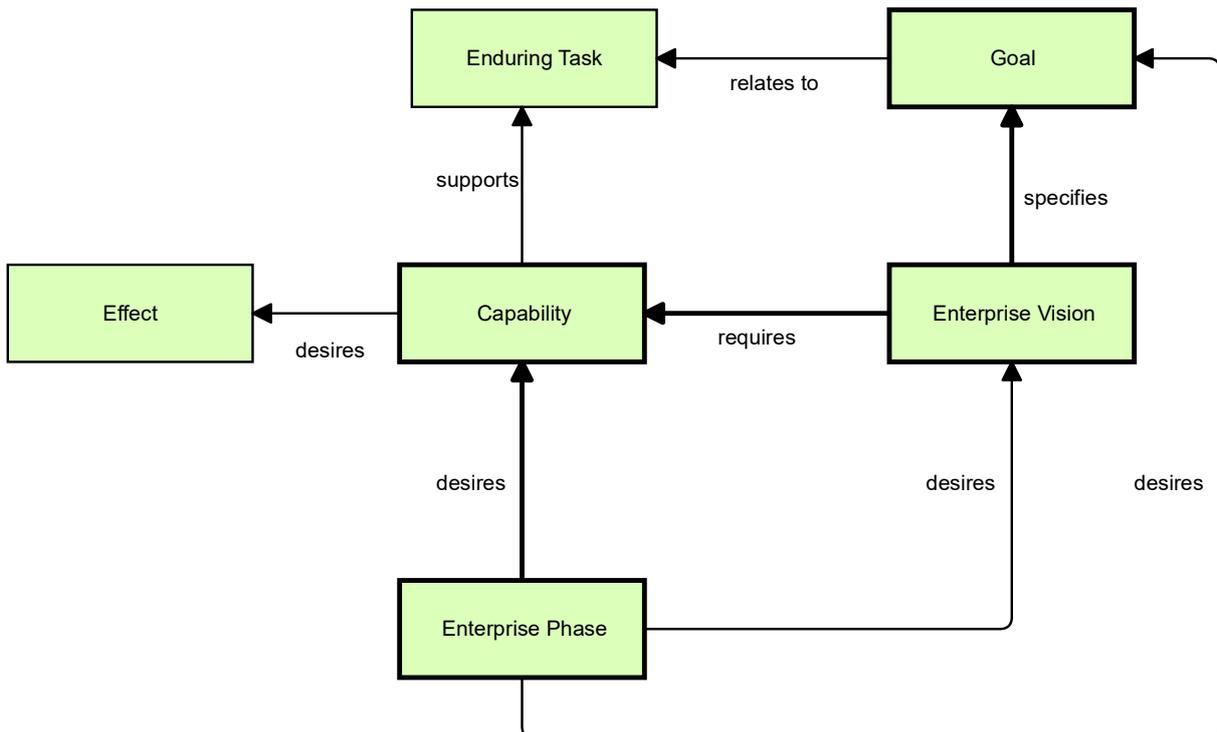
5 C2 - ENTERPRISE VISION

Purpose	Usage & Concerns Addressed
<p>The C2 Viewpoint is concerned with the scoping of the Architectural effort and providing the strategic context for the Capabilities described in the Architecture.</p>	<ul style="list-style-type: none"> • 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.

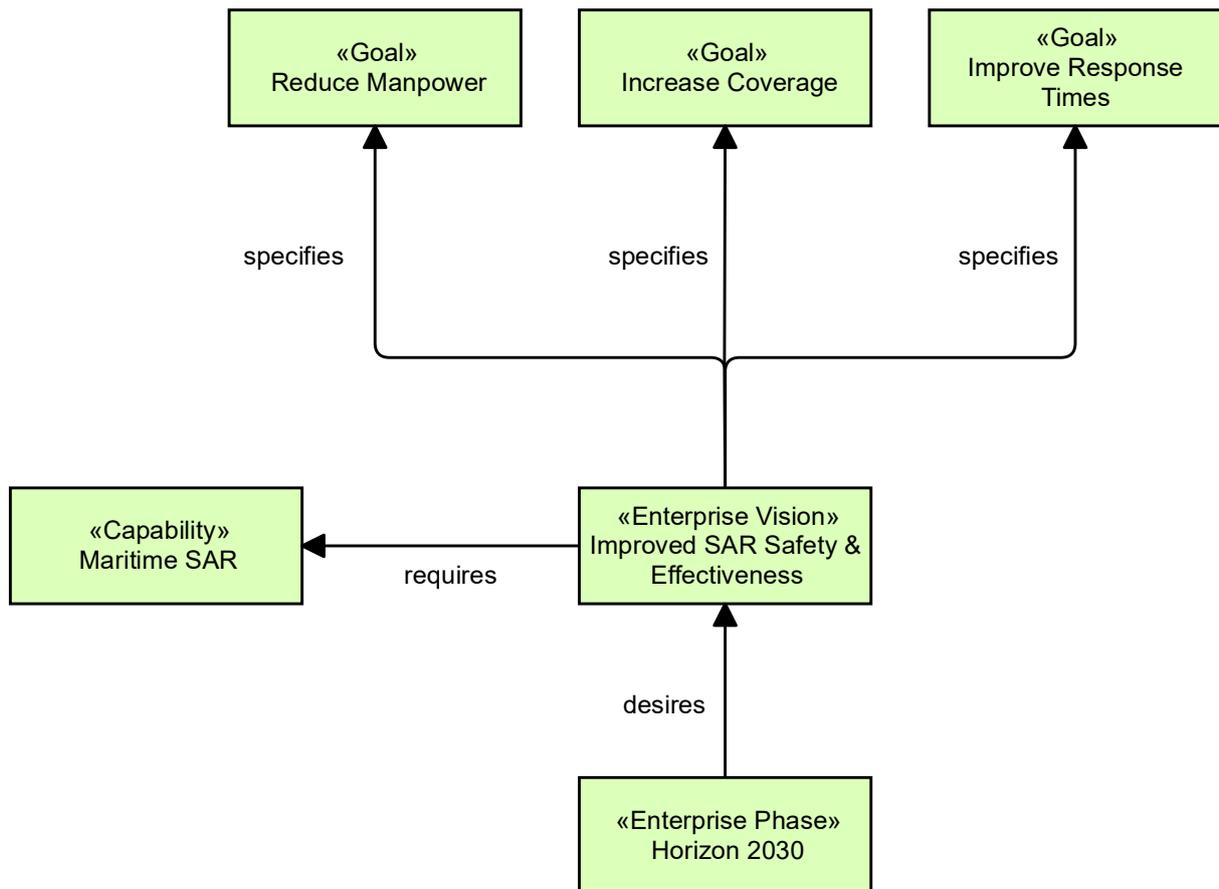
Scope
<ul style="list-style-type: none"> • Shall contain the Enterprise Vision and the required Capabilities. • Shall contain Goals specified by the Vision. • Shall contain Enterprise Phases and the Capabilities they desire. • May show the Effects that the Capabilities desire. • May show Enduring Tasks linked to Capabilities and Goals.

Representation
<ul style="list-style-type: none"> • Structured Text. • Composite Structure Diagram.

5.1 Cut-out



5.2 NAFv4 Compliant Example



6 C3 - CAPABILITY DEPENDENCIES

Purpose	Usage & Concerns Addressed
The C3 Viewpoint is concerned with the identification of dependencies between capabilities.	<ul style="list-style-type: none"> • Capability Management. • Analysis of dependencies between capabilities and between Capability groups. • 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.

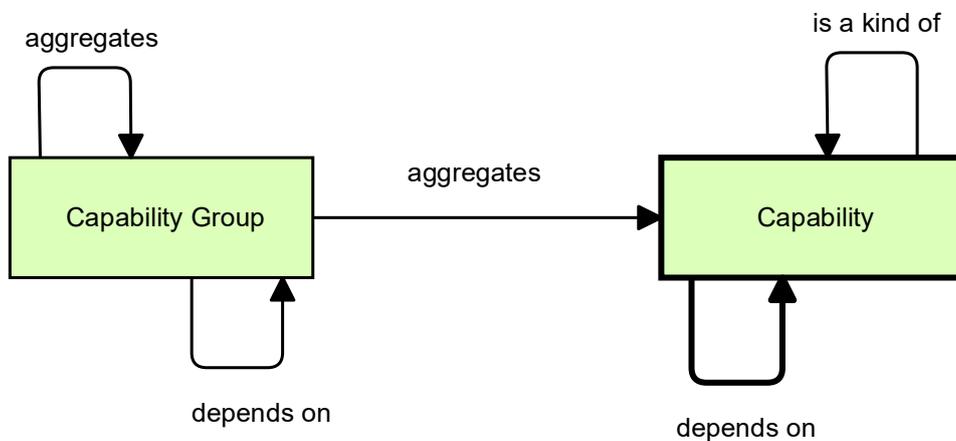
Scope

- Shall contain all dependencies between Capabilities relevant for the entity of interest.
- May show the Capabilities Groups that the Capabilities belong to.
- May show dependencies between Capability Groups.
- May show Capability specializations.

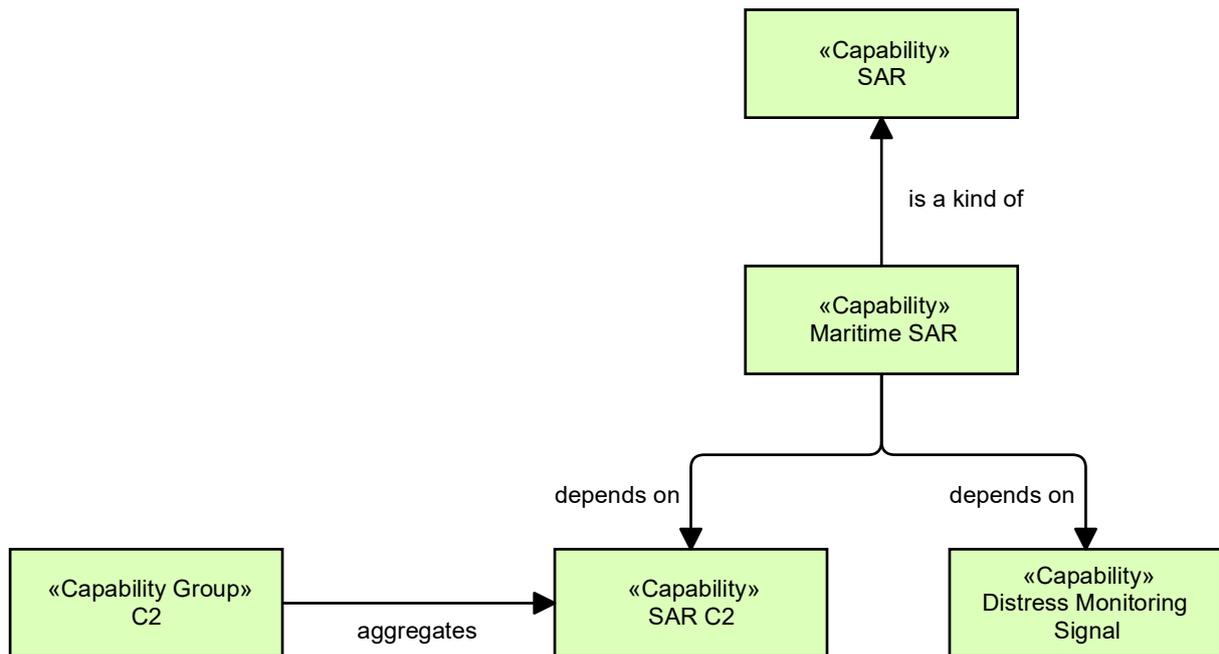
Representation

- 'Nested box' diagram.
- Class diagram.
- Composite Structure diagram.

6.1 Cut-out



6.2 NAFv4 Compliant Example



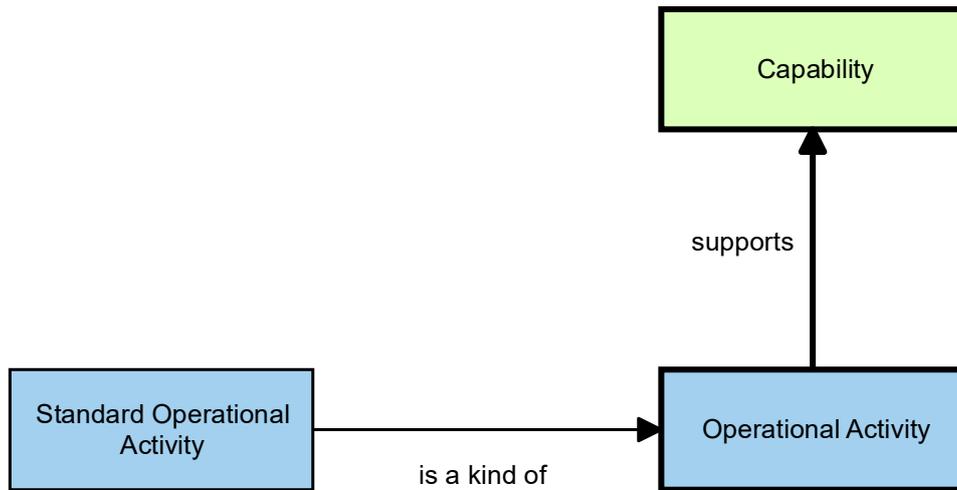
7 C4 - STANDARD PROCESSES

Purpose	Usage & Concerns Addressed
The C4 Viewpoint is concerned with the identification of processes with the traceability to the Capabilities they support.	<ul style="list-style-type: none"> • Doctrine Production. • Operational Analysis. • Specification of doctrine. • Tracing Capabilities to Standard Operational Activities. • Capability audit. • Concept of Operations.

Scope
<ul style="list-style-type: none"> • Shall contain Operational Activities relevant for the entity of interest. • Shall contain the capabilities that Operational Activities support. • May show Standard Operational Activities that support capabilities.

Representation
<ul style="list-style-type: none"> • Tabular. • Tracing Diagram.

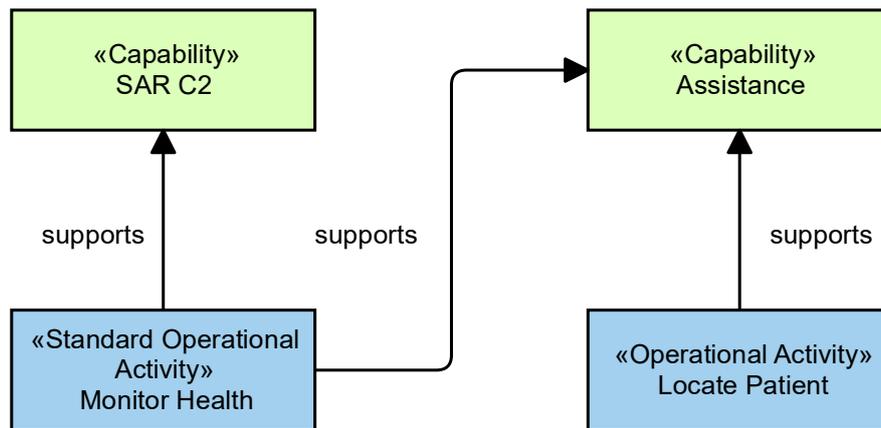
7.1 Cut-out



7.2 Notes on the Cut-out

Standard Operational Activities are Operational Activities that are specified by doctrine.

7.3 NAFv4 Compliant Example



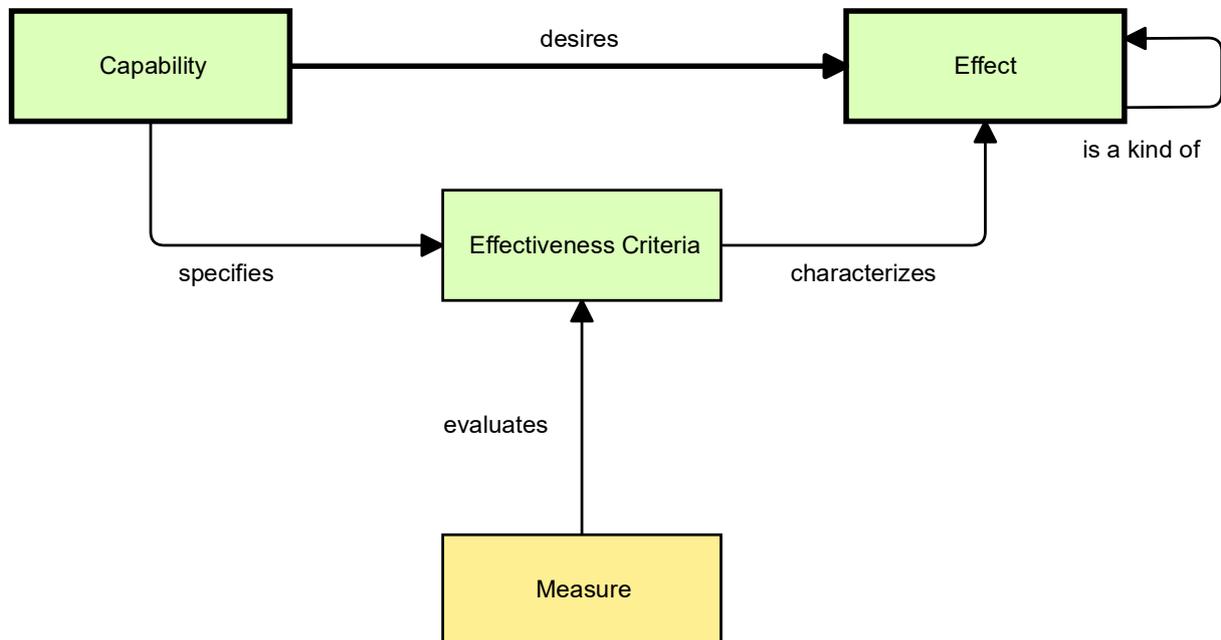
8 C5 - EFFECTS

Purpose	Usage & Concerns Addressed
<p>The C5 Viewpoint is concerned with identifying and describing desired effects of Capabilities and the Effectiveness Criteria that are specified for them.</p>	<ul style="list-style-type: none"> • Operational Analysis. • Analysis of non-functional properties. • Characterization of the expected results capabilities, noting that effects can be positive or negative. • Analysis of cumulative effects. • Analysis of persistence of the effects. • Tracing the operational states and modes with regards to the effects.

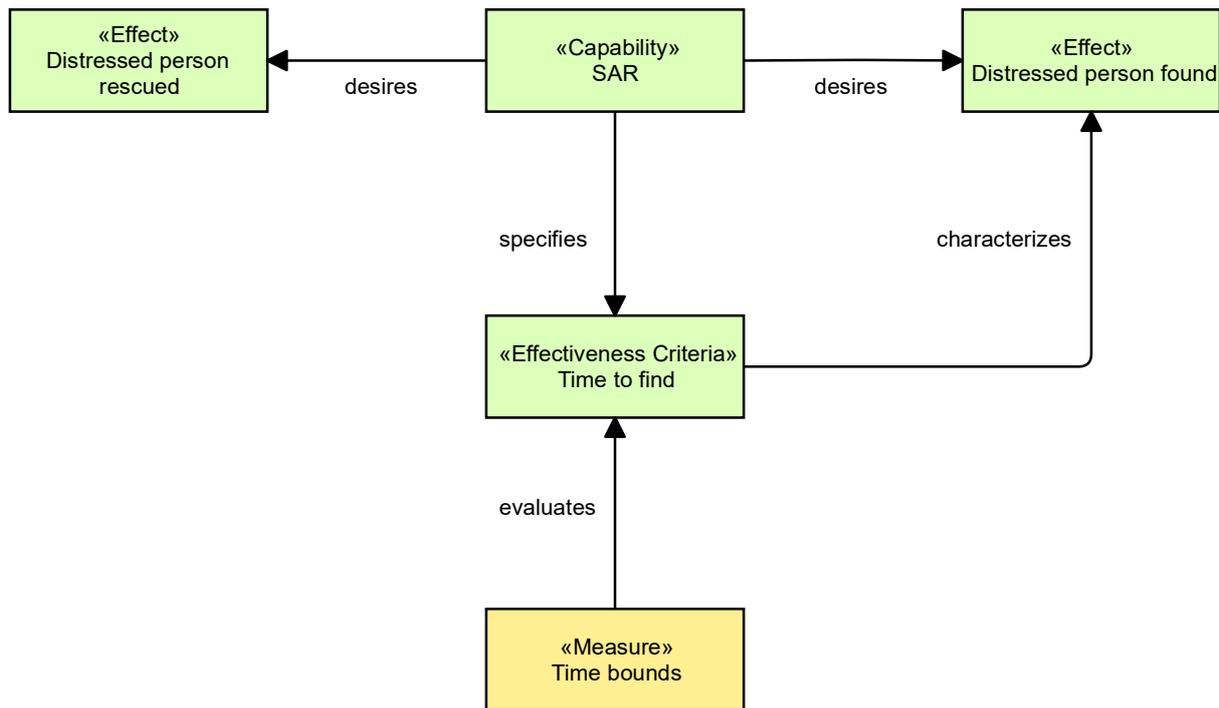
Scope
<ul style="list-style-type: none"> • Shall contain Effects that the Capabilities desire. • May show Effectiveness Criteria that is specified by Capabilities. • May show Effectiveness Criteria characterizing the Effects. • May show measures that evaluate the Effectiveness Criteria.

Representation
<ul style="list-style-type: none"> • Tabular. • Structural diagram.

8.1 Cut-out



8.2 NAFv4 Compliant Example



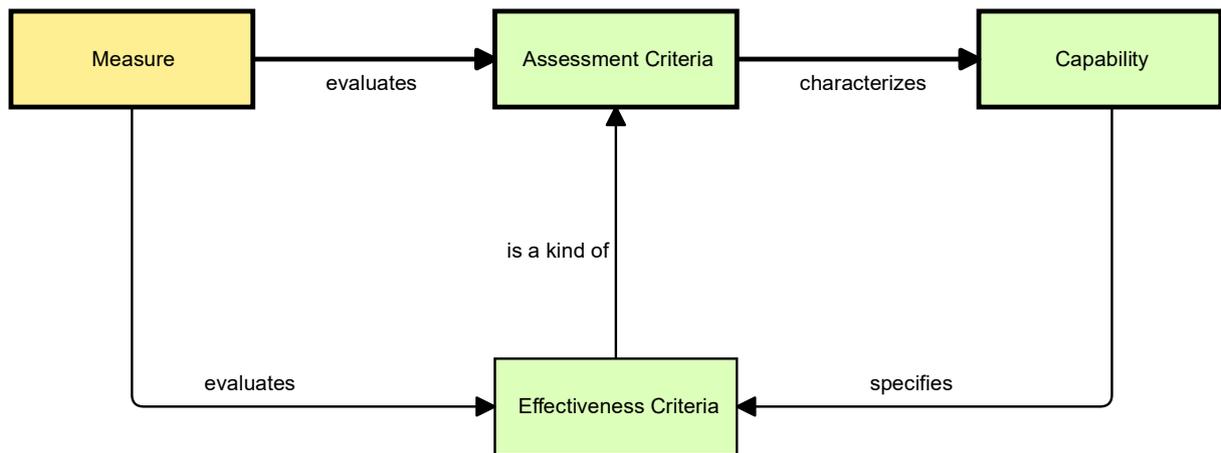
9 C7 - PERFORMANCE CRITERIA

Purpose	Usage & Concerns Addressed
The C7 Viewpoint is concerned with the identification and description of measure categories (Assessment Criteria) and identification of capabilities to which they are applicable.	<ul style="list-style-type: none"> • Capability Planning. • Capability Management. • Setting Capability Assessment Criteria. • Military Estimates. • Strategic Reviews. • Planning Assumptions.

Scope
<ul style="list-style-type: none"> • Shall contain the Capabilities characterized by the Assessment Criteria. • Shall contain Measures that evaluate Assessment Criteria. • May show Effectiveness Criteria specified by Capabilities. • May show Effectiveness Criteria that is evaluated by Measures.

Representation
<ul style="list-style-type: none"> • Tabular (capabilities on one axis, measure categories on the other). • Class diagram with property definitions.

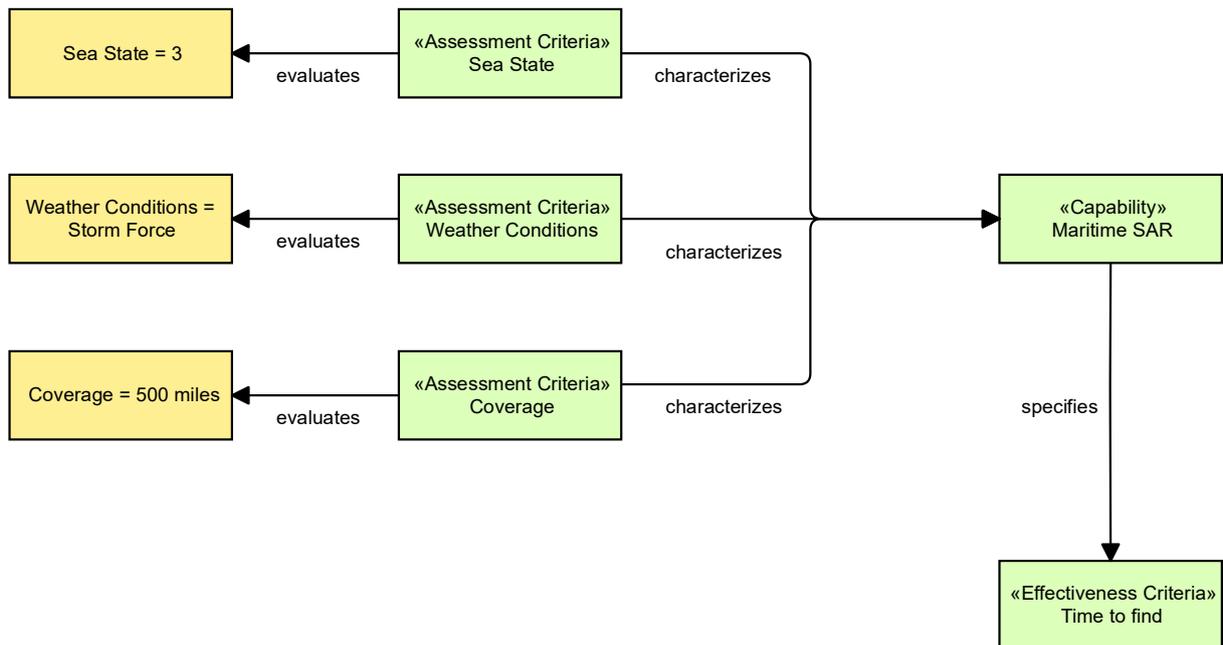
9.1 Cut-out



9.2 Notes on the Cut-out

The C7 provides the possibility to specify the full range of Assessment Criteria while the C5 is focusing on Effectiveness Criteria which are a special case of Assessment Criteria.

9.3 NAFv4 Compliant Example



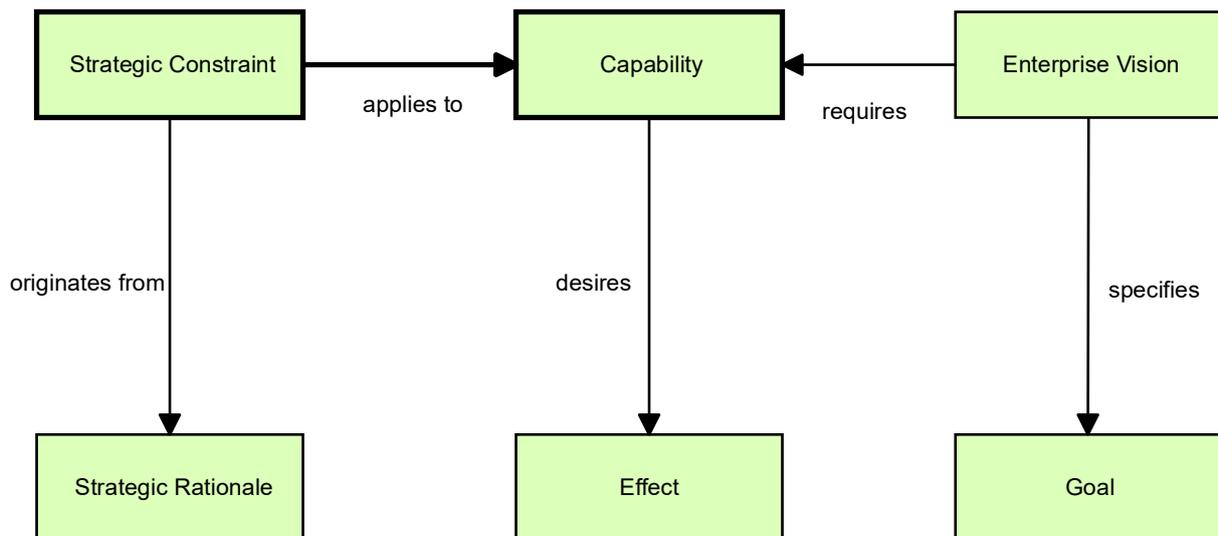
10 C8 - PLANNING CONSTRAINTS

Purpose	Usage & Concerns Addressed
The C8 Viewpoint is concerned with the identification and description of assumptions (Strategic Constraints).	<ul style="list-style-type: none"> • Capability Planning. • Planning Assumptions. • Implementation Planning.

Scope
<ul style="list-style-type: none"> • Shall contain Capabilities relevant for the entity of interest. • Shall contain Strategic Constraints that apply to the Capabilities. • May show Strategic Rationale relating to Strategic Constraints. • May show Effects the Capabilities desire. • May show the Vision that requires the Capability and the Goal it specifies.

Representation
<ul style="list-style-type: none"> • Tabular. • Benefits diagram. • Parametric diagram.

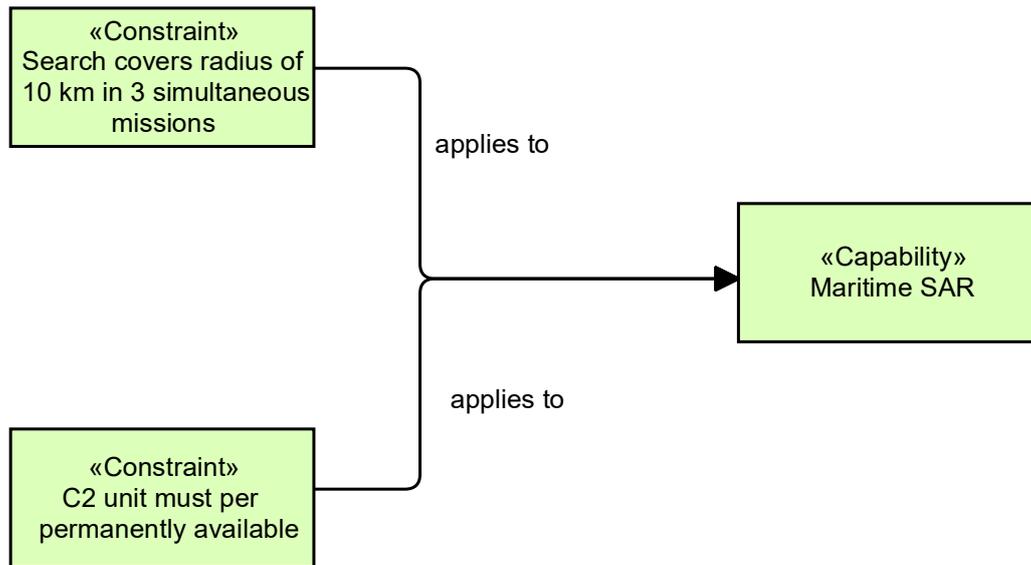
10.1 Cut-out



10.2 Notes on the Cut-out

Planning Assumptions, as defined in the NAFv3, are represented as a Strategic Constraint (recognising the fact that an assumption has a lesser level of rigour than a constraint.)

10.3 NAFv4 Compliant Example



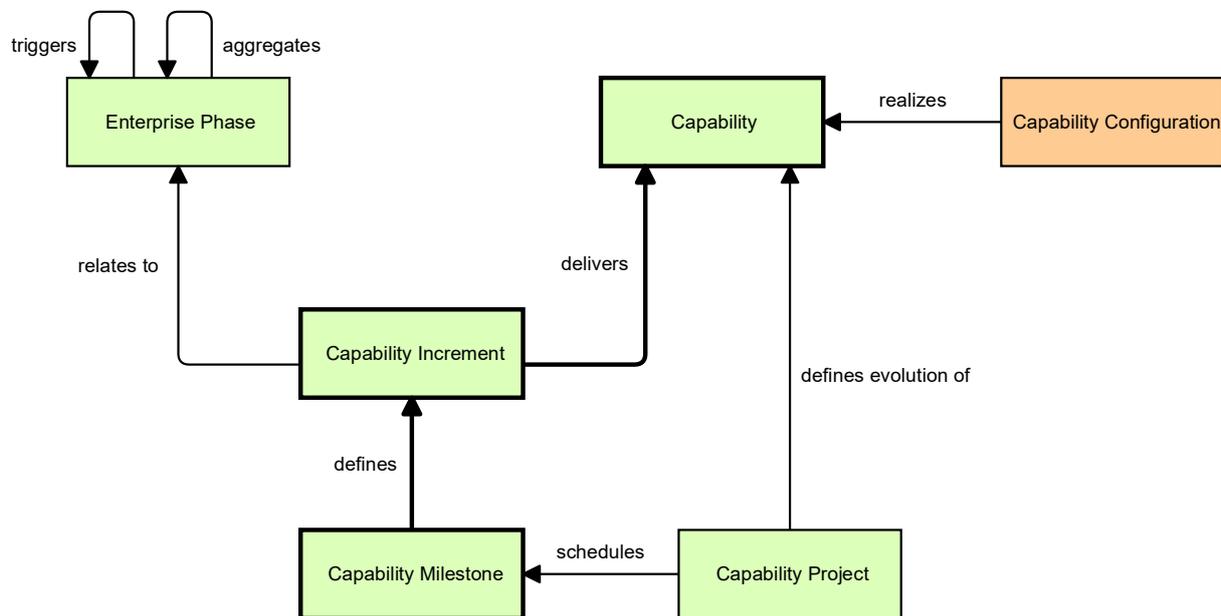
11 CR - CAPABILITY ROADMAP

Purpose	Usage & Concerns Addressed
The Cr Viewpoint is concerned with identifying Capability Milestones related to Capability Increments and the Capabilities it delivers.	<ul style="list-style-type: none"> • Capability Planning. • Acquisition Management. • Capability phasing. • Capability integration planning. • Capability gap/surplus analysis. • High-level dashboard for acquisition management.

Scope
<ul style="list-style-type: none"> • Shall contain Capability Milestones. • Shall contain the Capability Increments that the Capability Milestone defines. • Shall contain the Capabilities the Capability Increments delivers. • May show Enterprise Phases related to Capability Increments. • May show the ordering and grouping of Enterprise Phases. • May show Capability Projects and how they schedule Capability Milestones during the evolution of the Capability. • May trace Capability Configurations to Capabilities.

Representation
<ul style="list-style-type: none"> • A time based chart in the style of a Gantt chart.

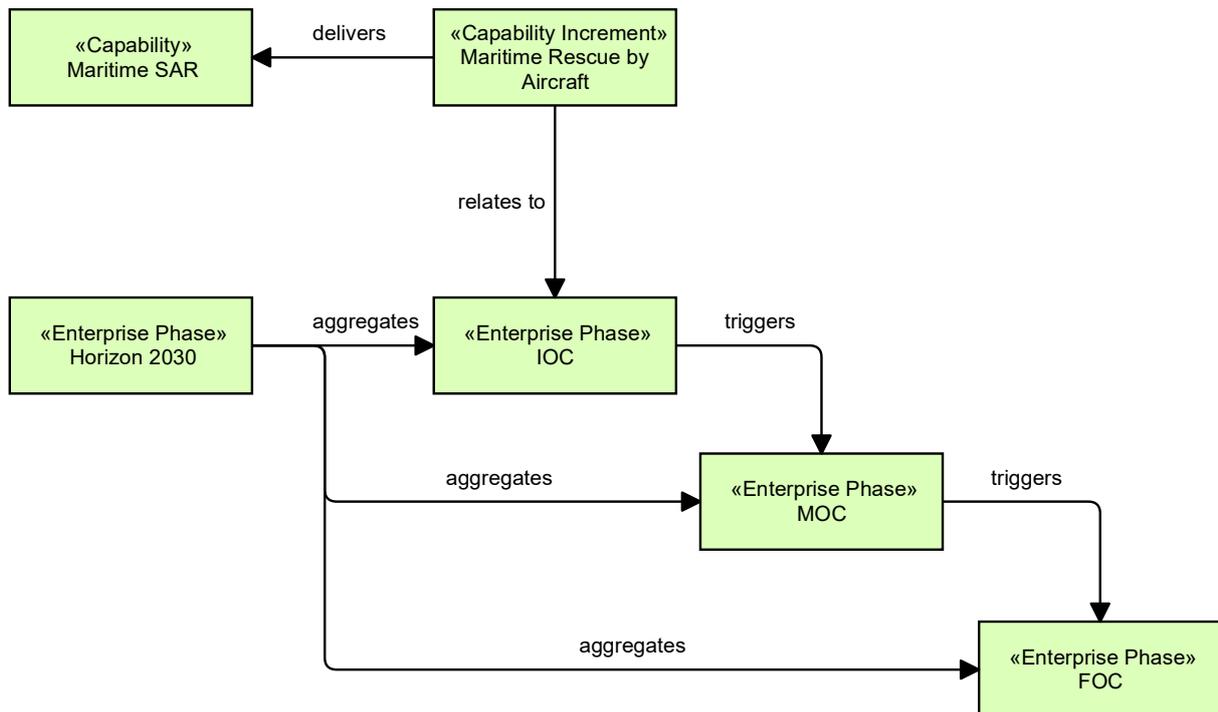
11.1 Cut-out



11.2 Notes on the Cut-out

Capability Configurations are included to allow for potential traceability to Capability.

11.3 NAFv4 Compliant Example



11.4 Notes on the Example

Milestones are not present in the example since they are properties of the capability increment. This example, similar to other roadmap viewpoints in NAFv4, is how to model the scenario and is important for the understanding of the evolution of the architecture over time and its impact on decisioning. However, the specific representation of the roadmap for the benefit of non-architectural stakeholder readership should be as described in the content of the 'Representation' section of this viewpoint definition.

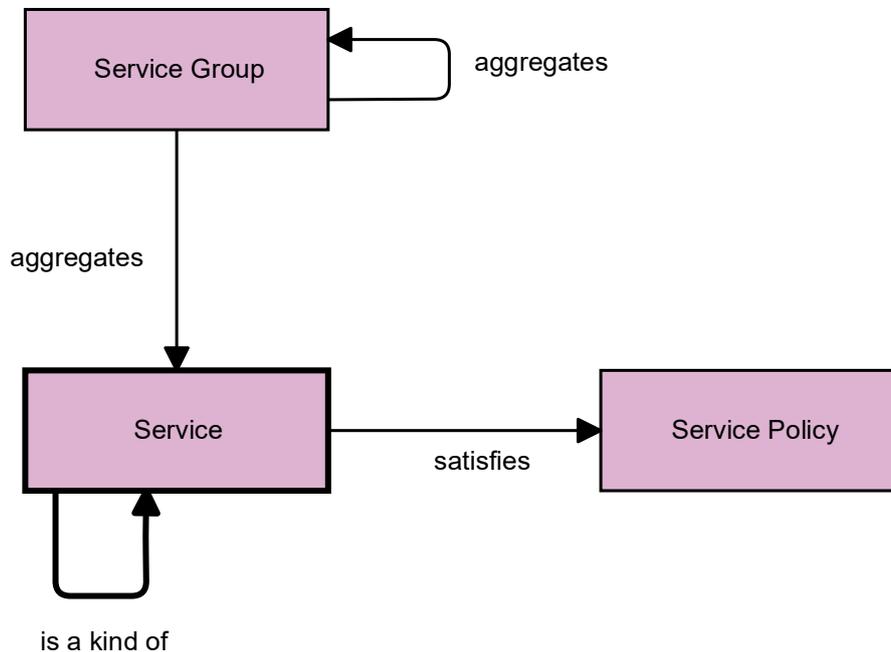
12 S1 - SERVICE TAXONOMY

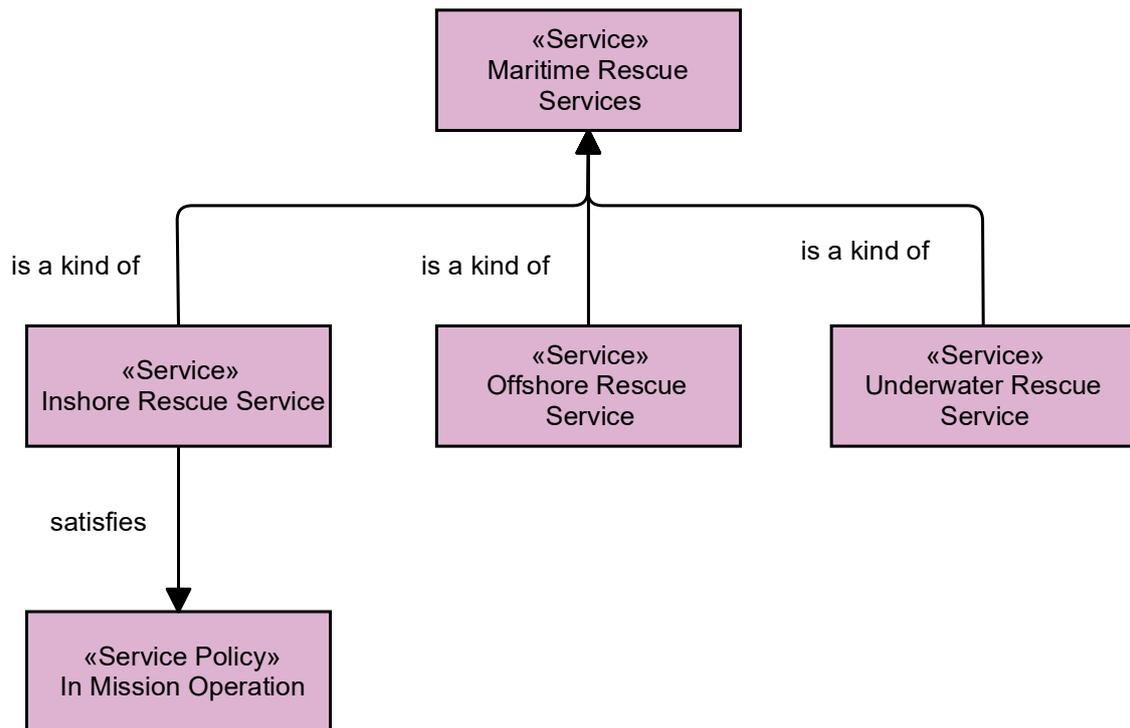
Purpose	Usage & Concerns Addressed
The S1 Viewpoint is concerned with the identification of service specifications, and their organization into specialization taxonomies.	<ul style="list-style-type: none"> • Service planning. • Service audit. • Identification of services. • Defining measures for Service Levels. • Service gap analysis. • Providing reference services for architectures. • Tailoring generic services for specific applications.

Scope
<ul style="list-style-type: none"> • Shall contain all Services organized into specialization taxonomies relevant for the entity of interest. • May show Services organized into Service Groups, and their groupings. • May show how Services satisfy Service Policies.

Representation
<ul style="list-style-type: none"> • Tabulation. • Hierarchical (connected shapes). • Class diagram.

12.1 Cut-out



12.2 NAFv4 Compliant Example

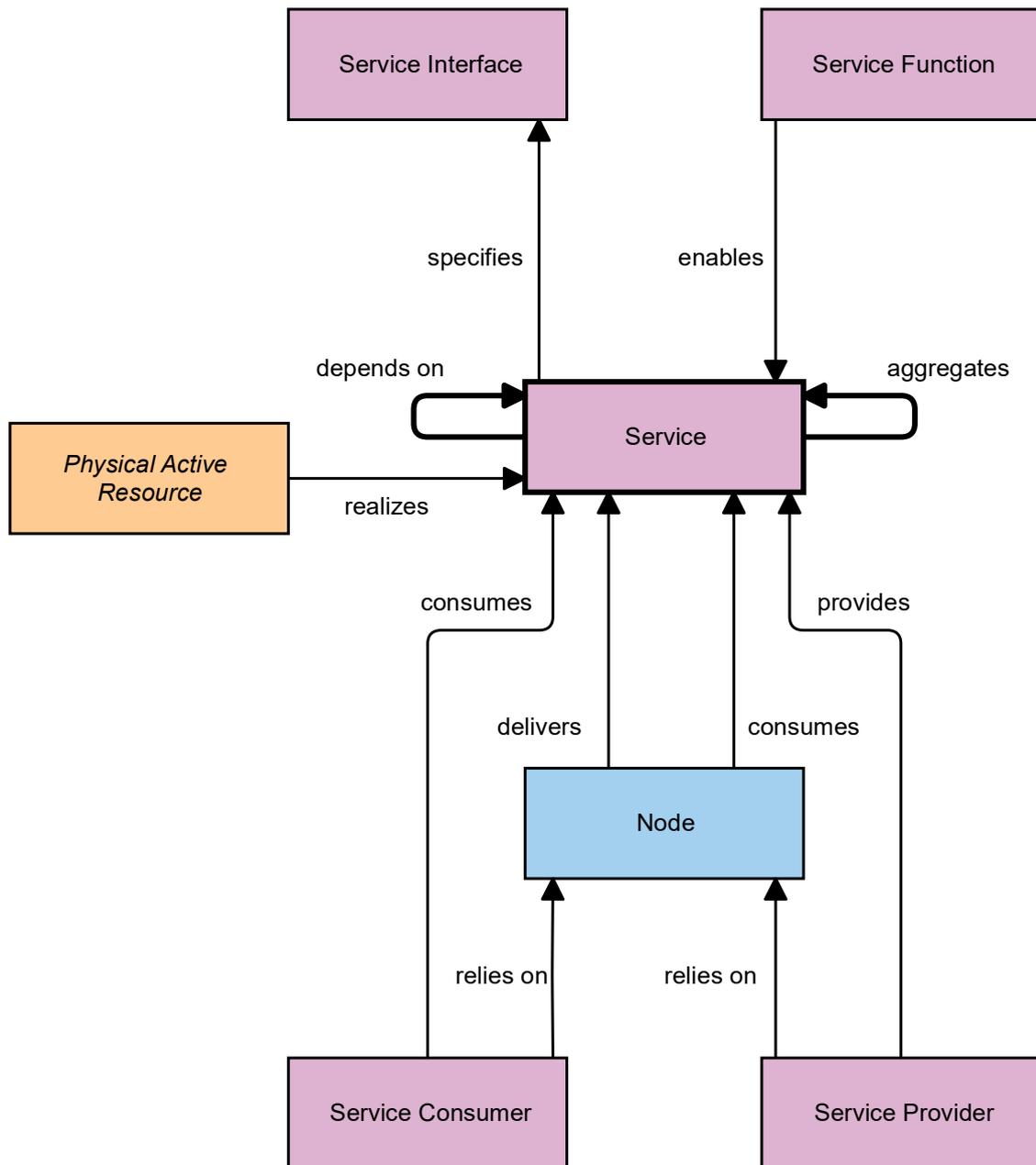
13 S2 - SERVICE STRUCTURE

Purpose	Usage & Concerns Addressed
The S2 Viewpoint is concerned with the identification of how Services are structured and their dependencies to other services.	<ul style="list-style-type: none"> • Service planning • Service composition. • Service dependency analysis. • Service interoperability.

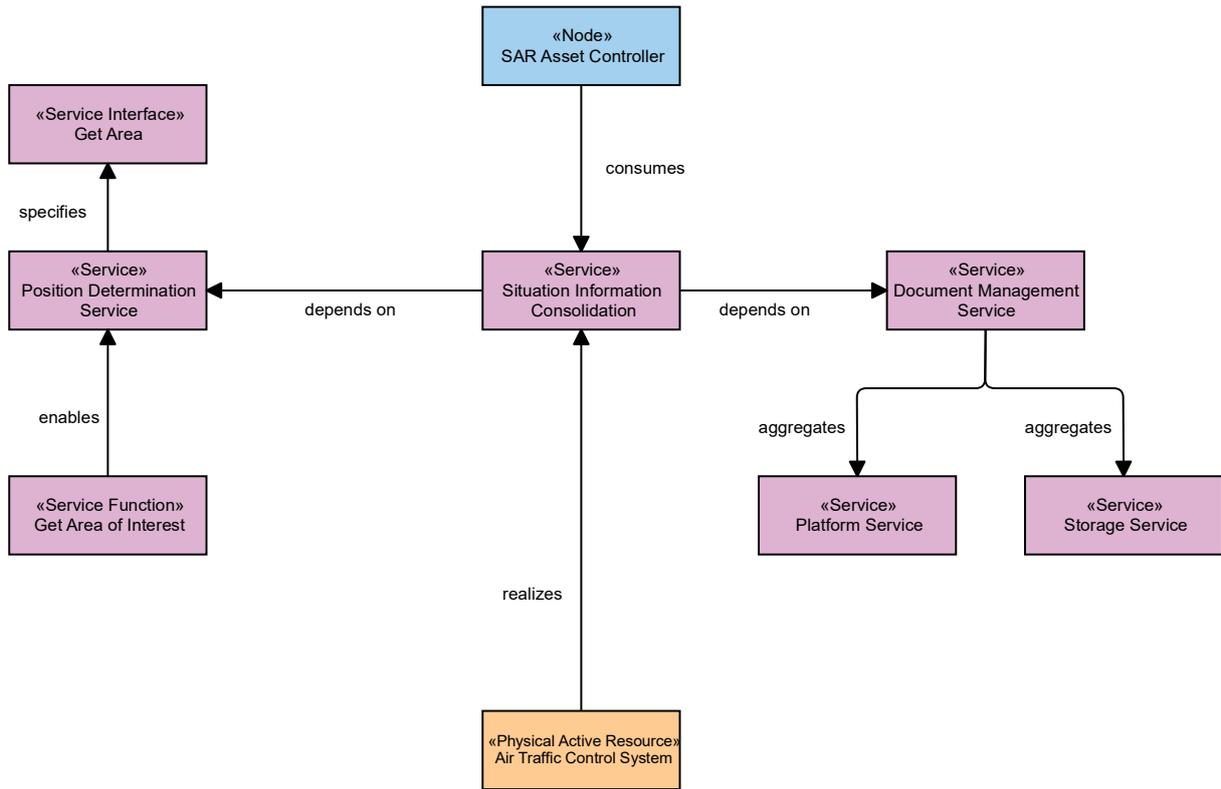
Scope
<ul style="list-style-type: none"> • Shall contain Services relevant for the entity of interest and their structural organization. • Shall contain dependencies between these Services. • May show Service Interfaces that the Service specifies. • May show Service Functions that the Service enables. • May show Service Consumers or Service Providers that consume/provide the Service. • May trace Services to Physical Active Resources. • May trace Services to Nodes that deliver or consume the Service and how these relate to a Service Provide or Service Consumer.

Representation
<ul style="list-style-type: none"> • Tabular. • Matrix. • Dependency graph. • Diagram.

13.1 Cut-out



13.2 NAFv4 Compliant Example



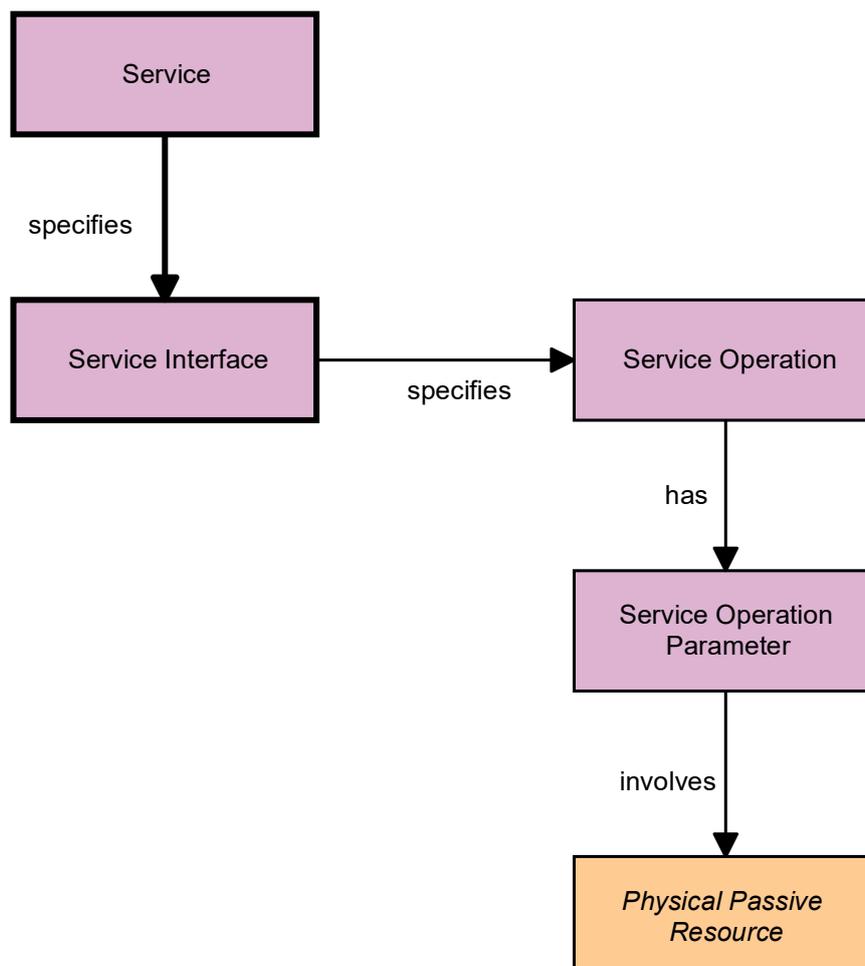
14 S3 - SERVICE INTERFACES

Purpose	Usage & Concerns Addressed
The S3 Viewpoint is concerned with the identification and specification of Service Interfaces.	<ul style="list-style-type: none"> • Service implementation guidance. • Service interoperability. • Service integration. • Defining Service interfaces and their boundaries.

Scope
<ul style="list-style-type: none"> • Shall contains Service Interfaces specified by a Service. • May show Service Operations for Service Interfaces. • May show Service Operation Parameters for Service Operations. • May trace Service Operation Parameters to Physical Passive Resources.

Representation
<ul style="list-style-type: none"> • Tabular. • Diagram.

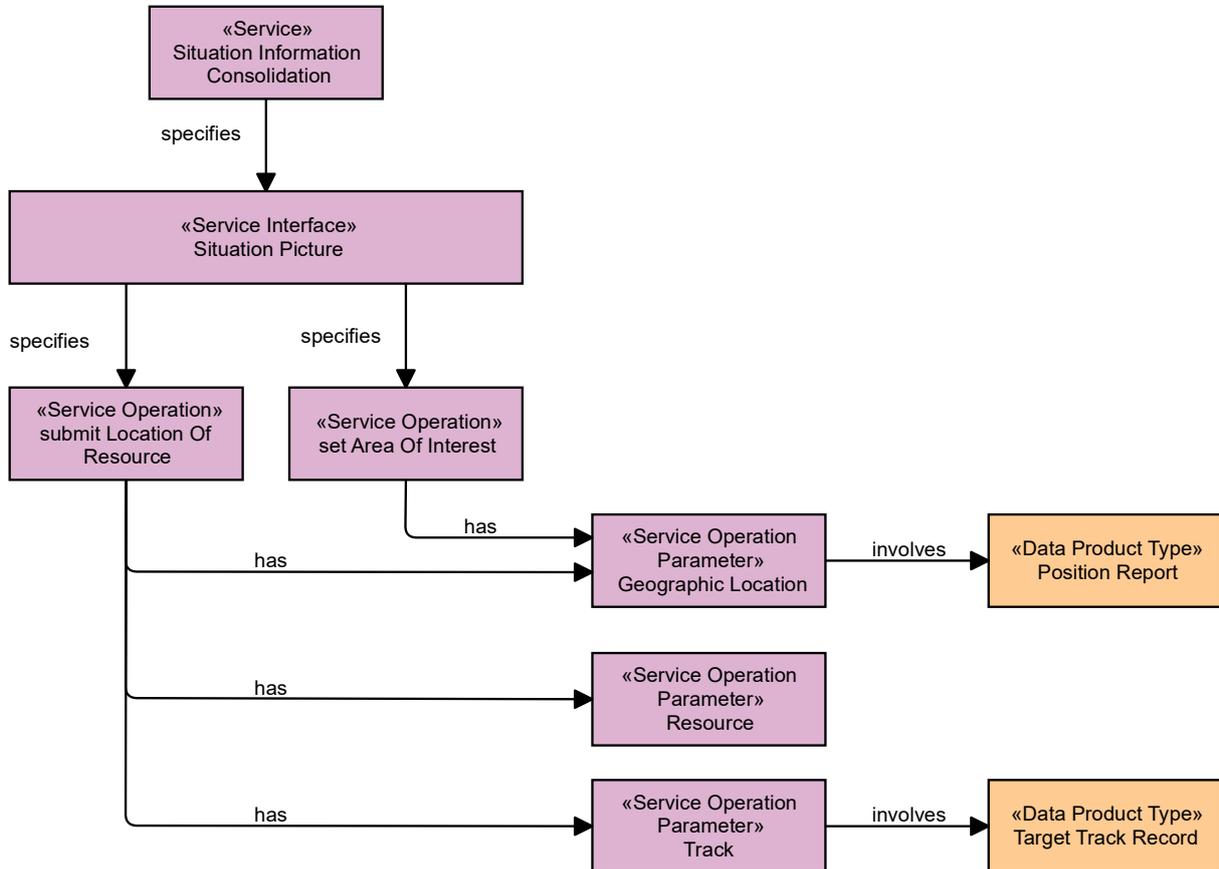
14.1 Cut-out



14.2 Notes on the Cut-out

The link between a Service Operation Parameter and a Data Model is realized transitively through a Physical Passive Resource.

14.3 NAFv4 Compliant Example



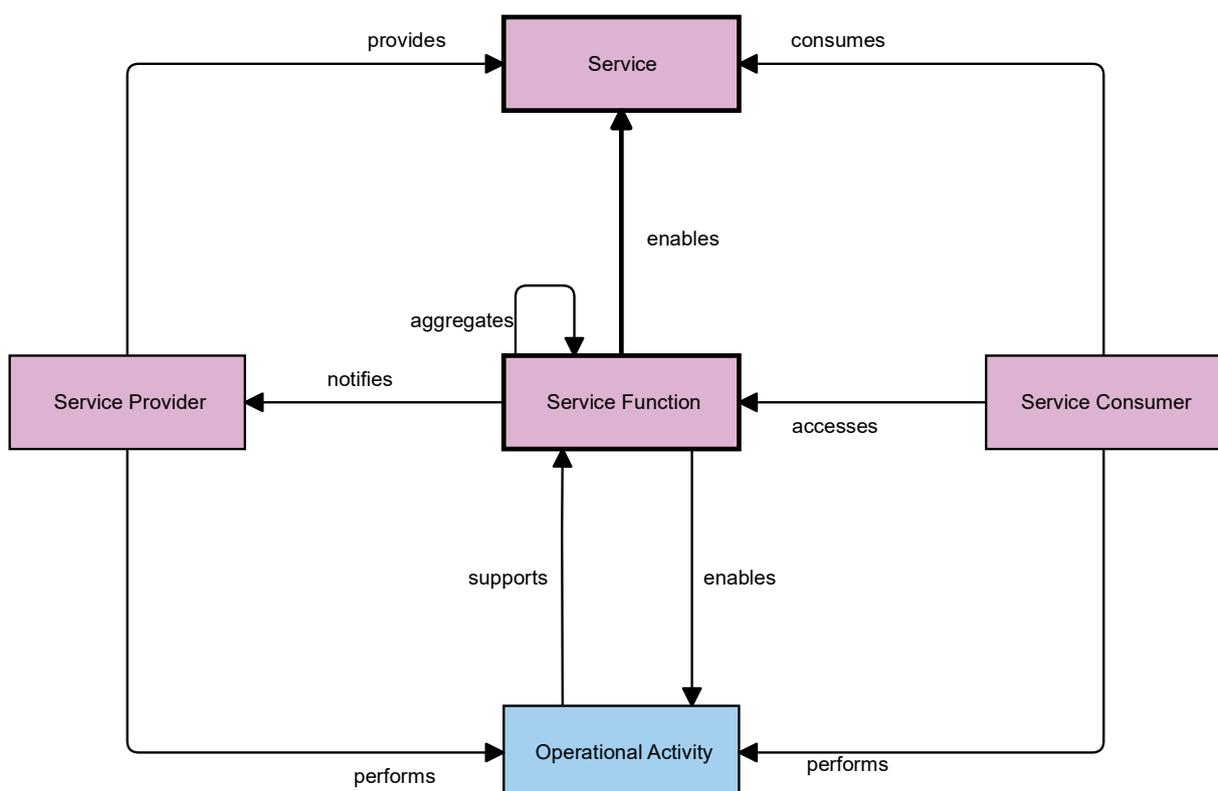
15 S4 - SERVICE FUNCTIONS

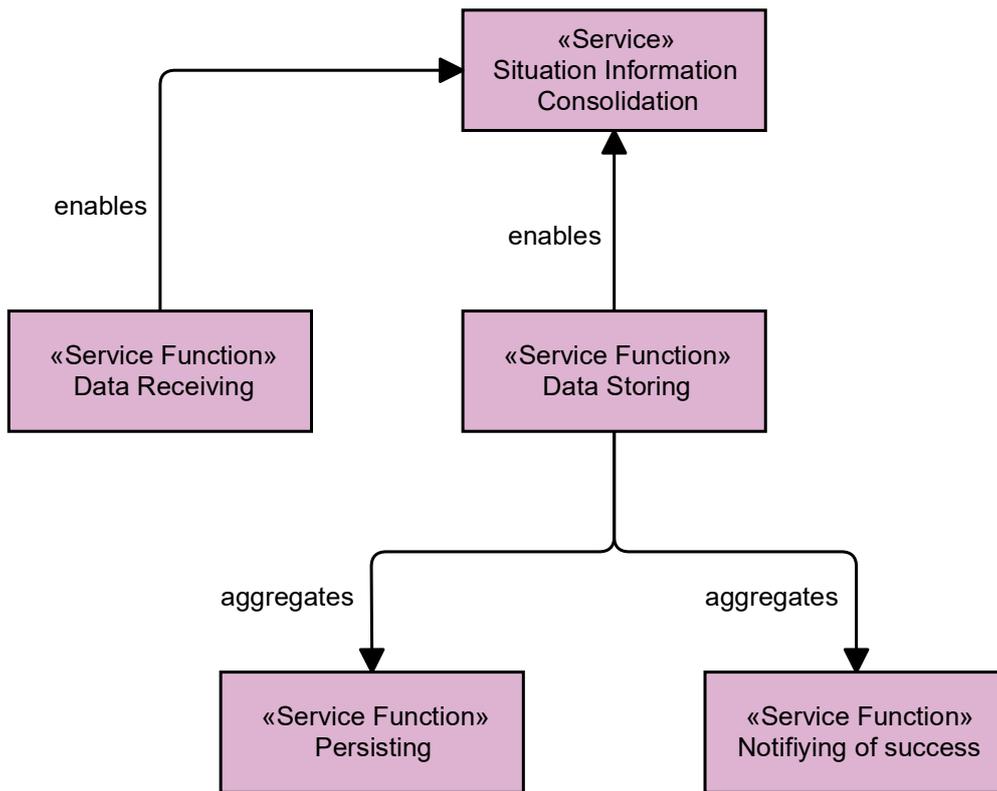
Purpose	Usage & Concerns Addressed
The S4 Viewpoint is concerned with the definition of the behaviour of a Service in terms of the Functions and Operational Activities that enable it.	<ul style="list-style-type: none"> • Better understanding of the level of detail required by a Service. • Outline requirements for Service behaviour. • Service implementation guidance. • Service specification & planning.

Scope
<ul style="list-style-type: none"> • Shall contain all Service Functions that enable a Service. • May show groupings of Service Functions. • May show Service Providers and/or Service Consumers relating to the Service Function or Service. • May show Operational Activities that support or are enabled by Service Functions. • May show Operational Activities that need to be performed by a Service Provider or Service Consumer.

Representation
<ul style="list-style-type: none"> • Tabular. • Diagram.

15.1 Cut-out

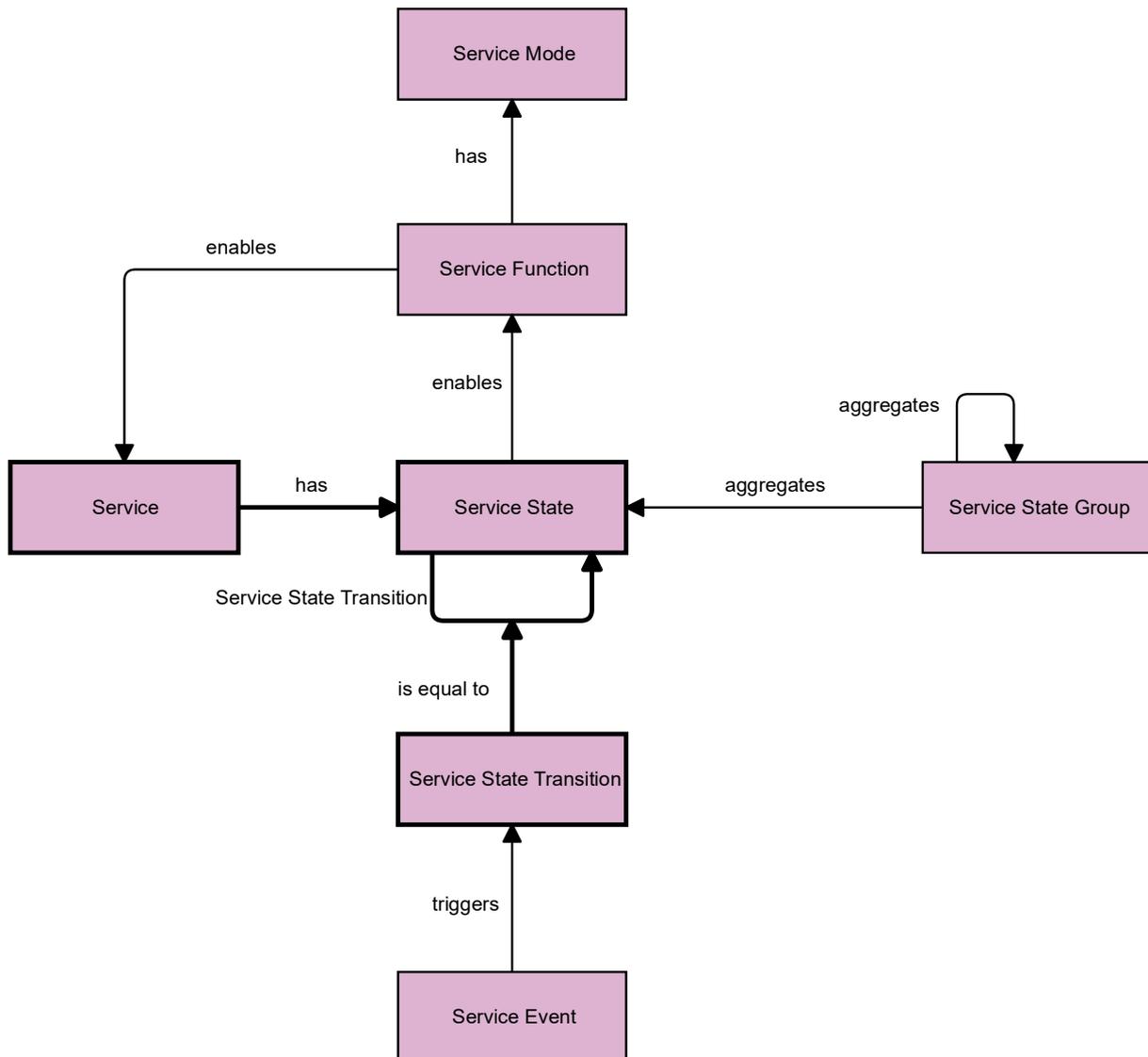


15.2 NAFv4 Compliant Example

16 S5 - SERVICE STATES

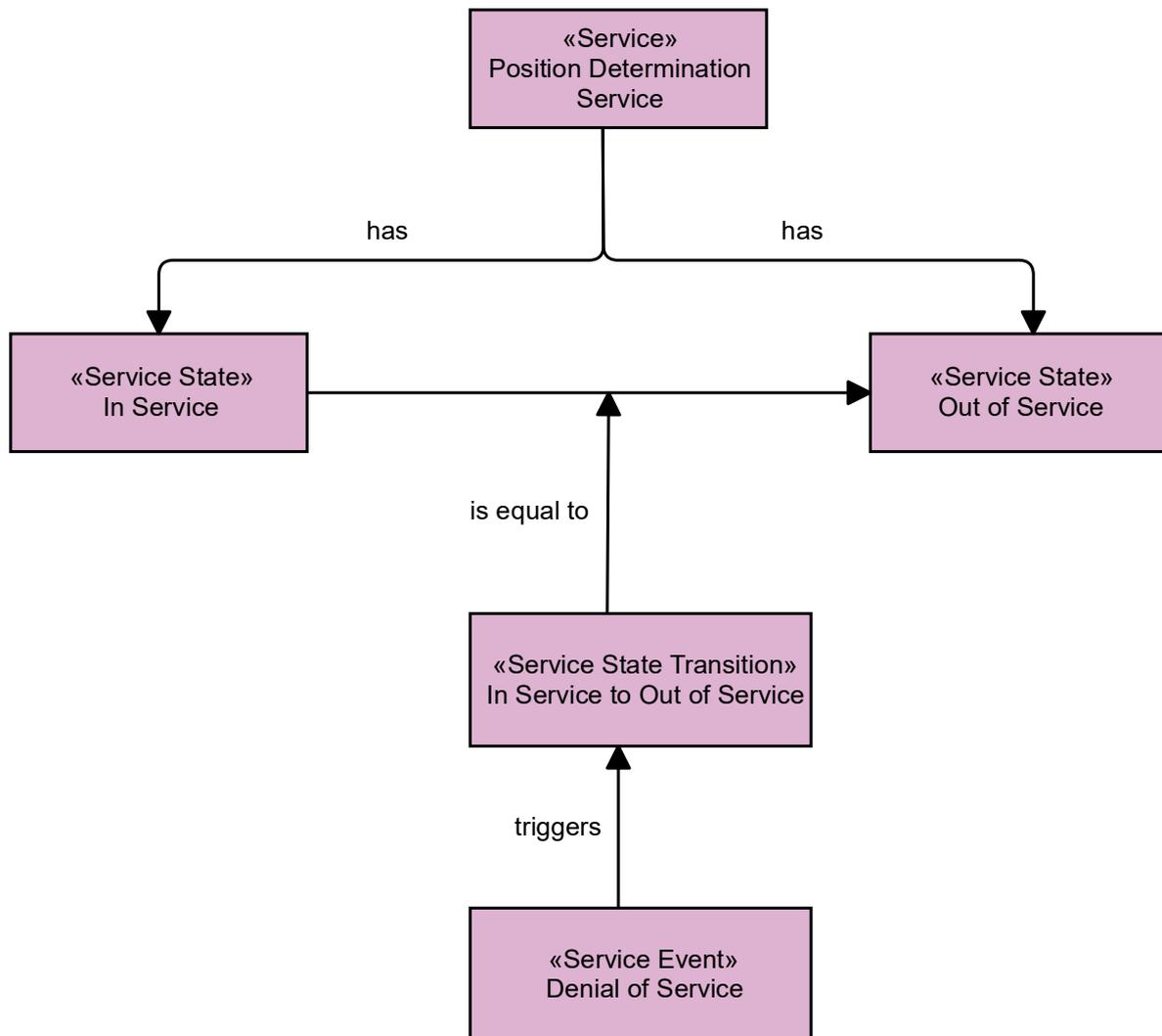
Purpose	Usage & Concerns Addressed
<p>The S5 Viewpoint is concerned with the identification and definition of the possible states a Service may have, and the possible transitions between those states.</p>	<ul style="list-style-type: none"> • Better understanding of the level of detail required by a Service. • Outline requirements for Service behaviour. • Service implementation guidance. • Service behaviour specification.
Scope	
<ul style="list-style-type: none"> • Shall contain all Service States of a Service relevant for the entity of interest. • Shall contain Service State Transitions between Service States. • May show Service Events triggering Service State Transitions. • May show Service Functions that are enabled by a Service State. • May show Service Functions that enable the Service. • May show Service Modes of a Service Function. • May show Service States grouped into Service State Groups, and their groupings. 	
Representation	
<ul style="list-style-type: none"> • Diagram. • State transition model. 	

16.1 Cut-out



16.2 Notes on the Cut-out

The *is equal to* relationship illustrates an **objectified relation**, where the element represents the instantiated form of a connection between other elements. This construct allows the relation itself to have further relationships providing a clearer and more flexible means of expressing complex interdependencies.

16.3 NAFv4 Compliant Example

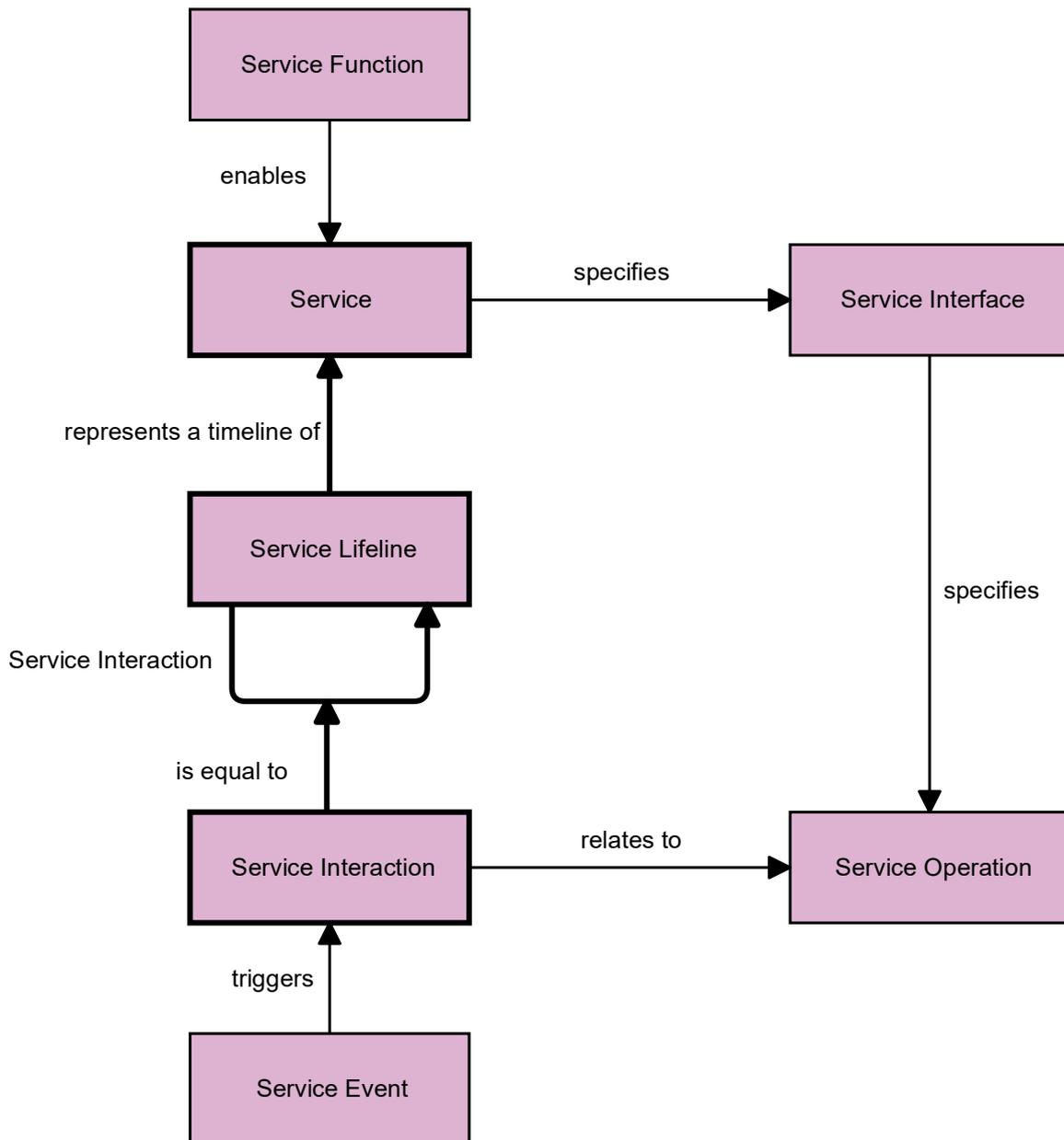
17 S6 - SERVICE SEQUENCE

Purpose	Usage & Concerns Addressed
The S6 Viewpoint is concerned with identifying the chronological sequence of interactions between services in a scenario.	<ul style="list-style-type: none">• Outline requirements for Service behaviour.• Service implementation guidance.• Service Orchestration.

Scope
<ul style="list-style-type: none">• Shall contain Service Lifelines representing Services.• Shall contain the Service Interactions between Service Lifelines.• May show Service Events that trigger Service Interactions.• May show Service Functions that enable the Service.• May show Service Interfaces specified by the Service.• May show Service Operations specified by the Service Interface or relating to the Service Interaction.

Representation
<ul style="list-style-type: none">• Sequence Diagram.

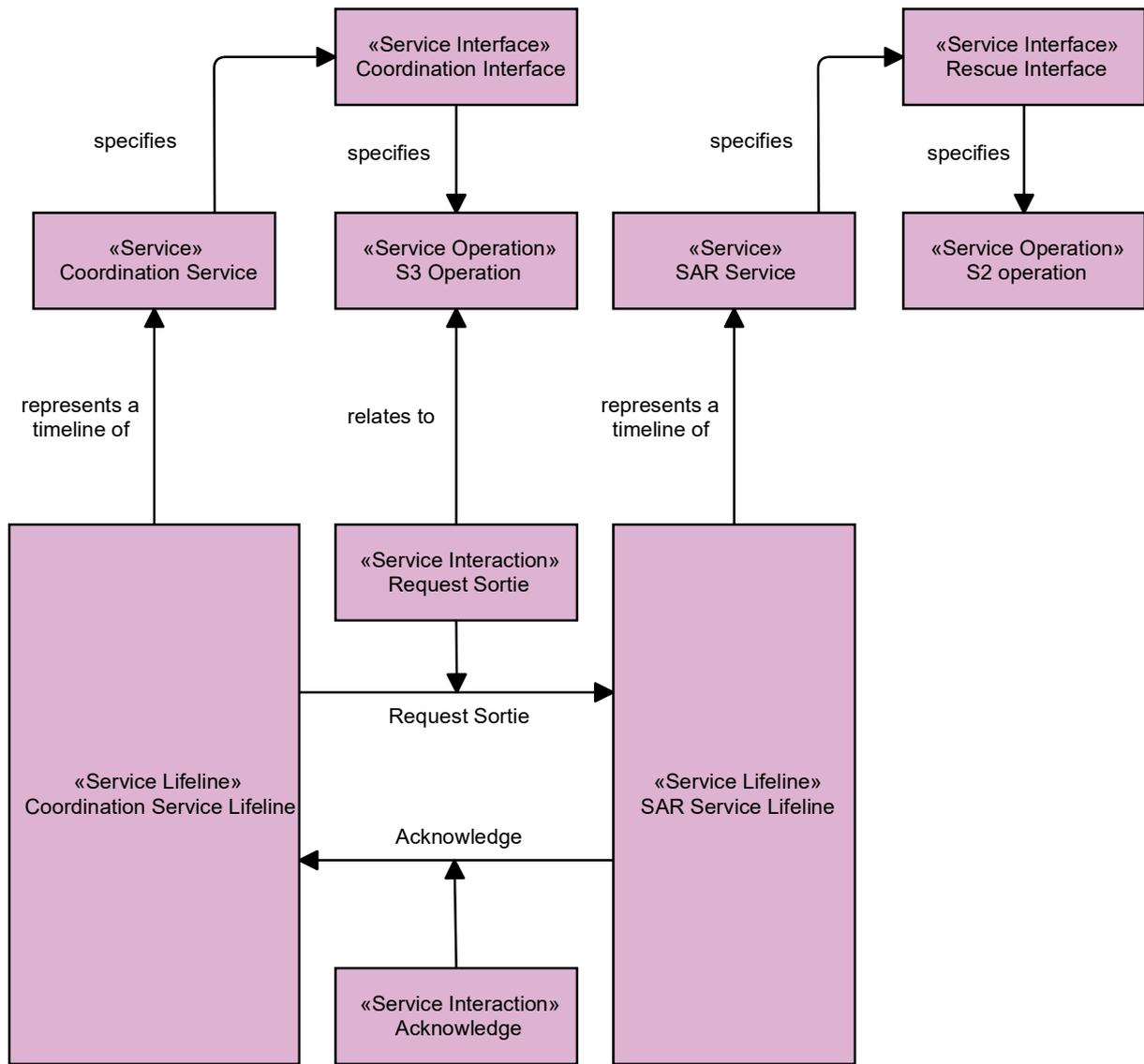
17.1 Cut-out



17.2 Notes on the Cut-out

The *is equal to* relationship illustrates an **objectified relation**, where the element represents the instantiated form of a connection between other elements. This construct allows the relation itself to have further relationships providing a clearer and more flexible means of expressing complex interdependencies.

17.3 NAFv4 Compliant Example



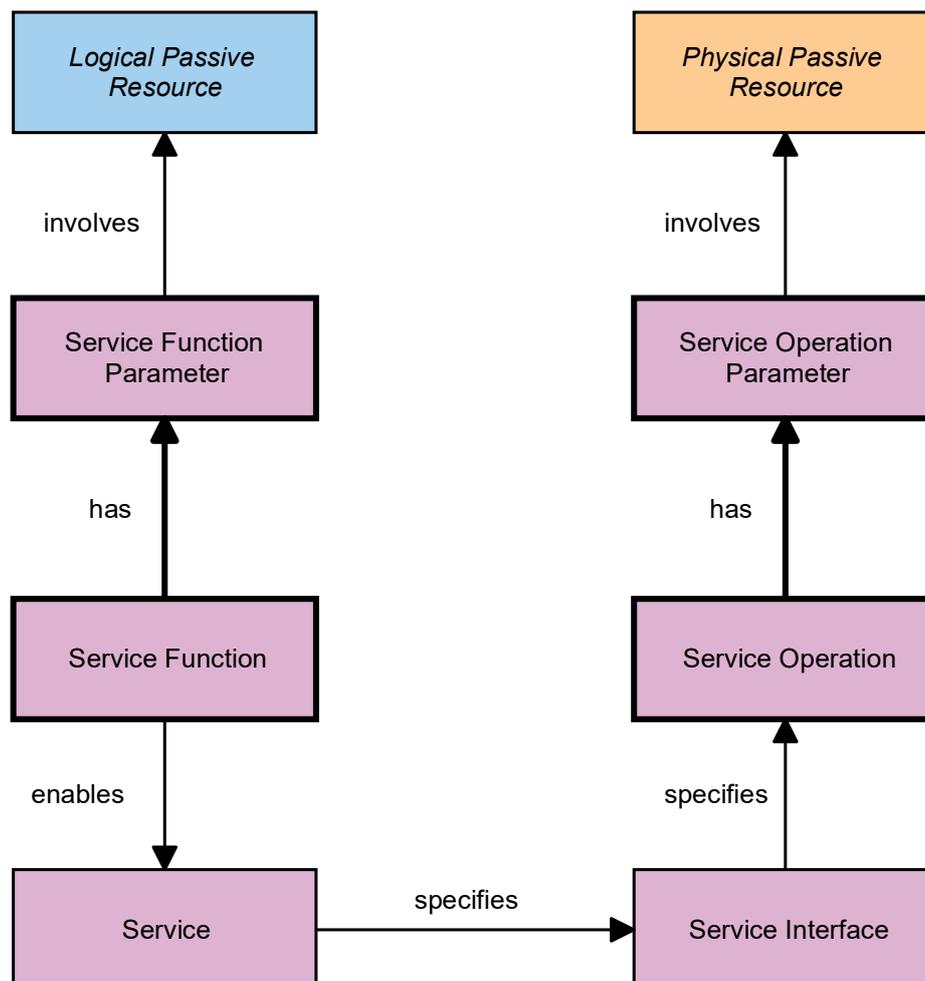
18 S7 SERVICE INTERFACE PARAMETERS

Purpose	Usage & Concerns Addressed
The S7 Viewpoint is concerned with the identification and specification of all the parameters used in service operations/ functions for the interface in scope.	<ul style="list-style-type: none"> • Detailed Service design. • Service compatibility analysis. • Service-oriented architecture governance. • Service interoperability.

Scope
<ul style="list-style-type: none"> • Shall contain parameters of Service Operations or parameters of Service Functions relevant for the entity of interest. • Shall contain Passive Resources involved in Services Functions or Service Operation Parameters. • May show Services that are enabled by the Service Functions. • May show Service Interfaces that specify the Service Operation, and the Services that specify the Service Interface.

Representation
<ul style="list-style-type: none"> • Tabular. • Diagram.

18.1 Cut-out

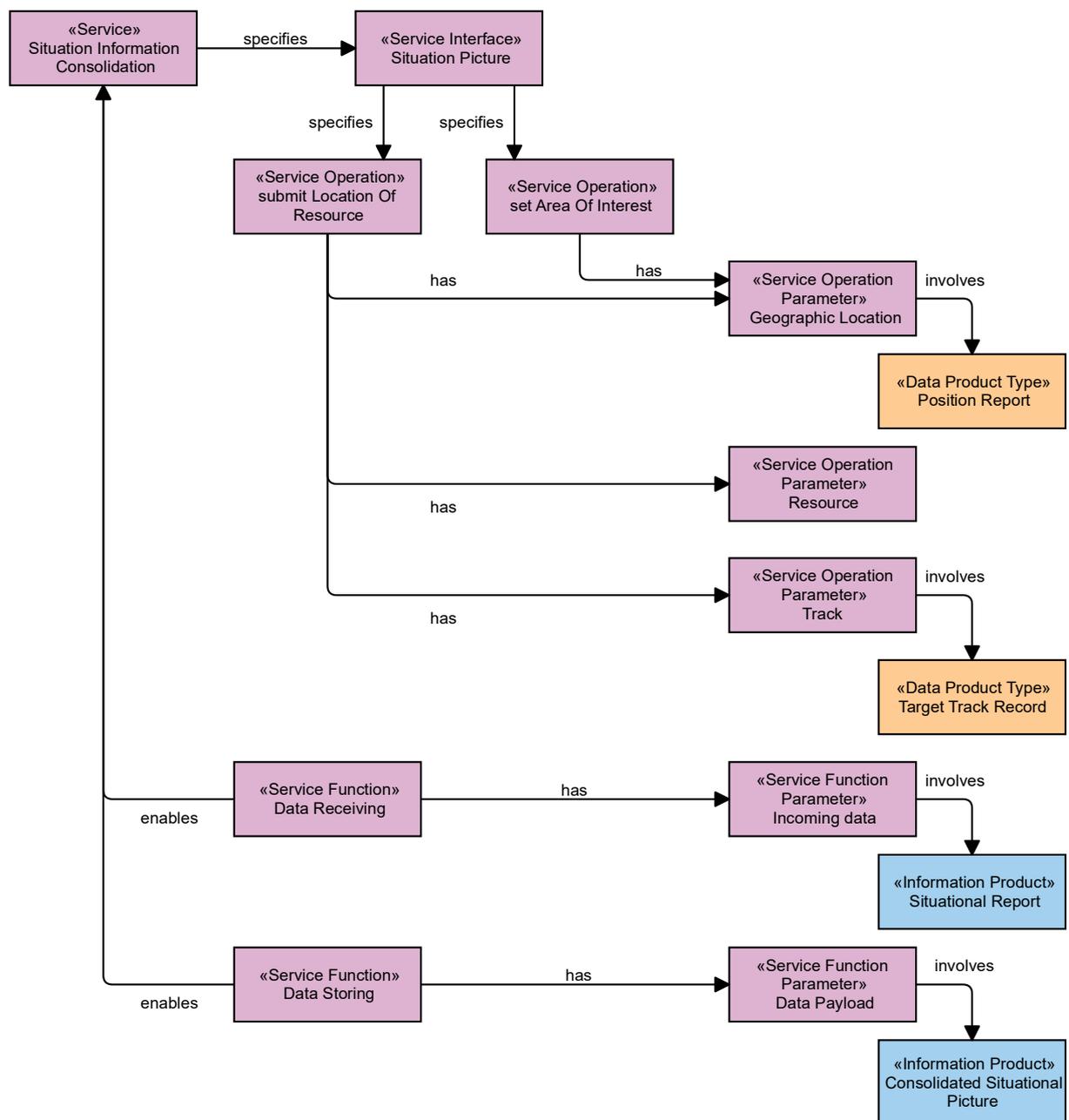


18.2 Notes on the Cut-out

Here, there are two ways of expressing service behaviour. *Service Functions* are typically used to describe non-technological service interactions, while *Service Interfaces* and their *Service Operations* usually denote the technological realization. In practice, both styles can be combined — for example, *Service Functions* may capture the problem space, whereas *Service Interfaces* and *Operations* describe the solution space.

The term *Service Parameter* is intentionally avoided to prevent confusion with SLAs or *Service Constraints*. Instead, parameters are distinguished as *Service Function Parameters* and *Service Operation Parameters*.

18.3 NAFv4 Compliant Example



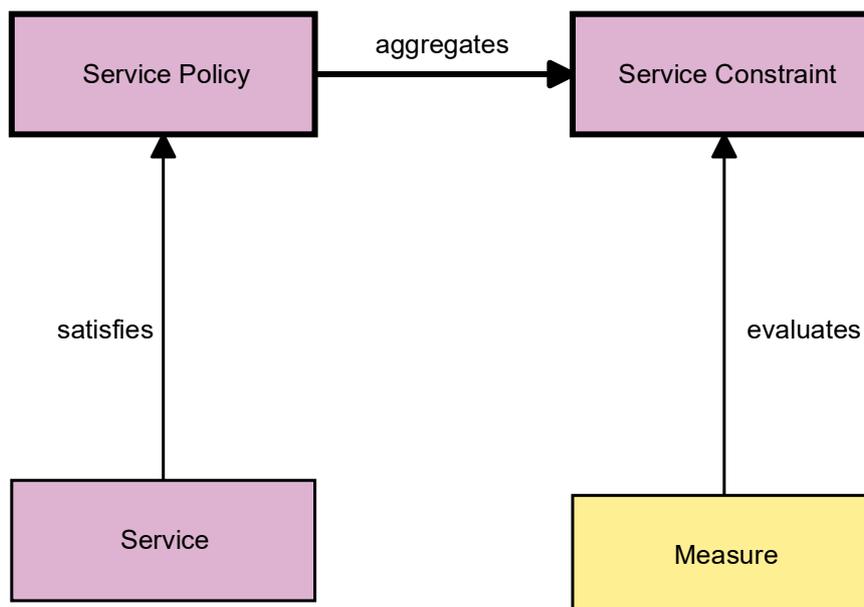
19 S8 - SERVICE CONSTRAINTS

Purpose	Usage & Concerns Addressed
The S8 Viewpoint is concerned with the identification and description of rules and constraints that apply to Service implementations, their operation and their use.	<ul style="list-style-type: none"> • Service design. • Contracting for Services. • User / System Requirements.. • Service governance.

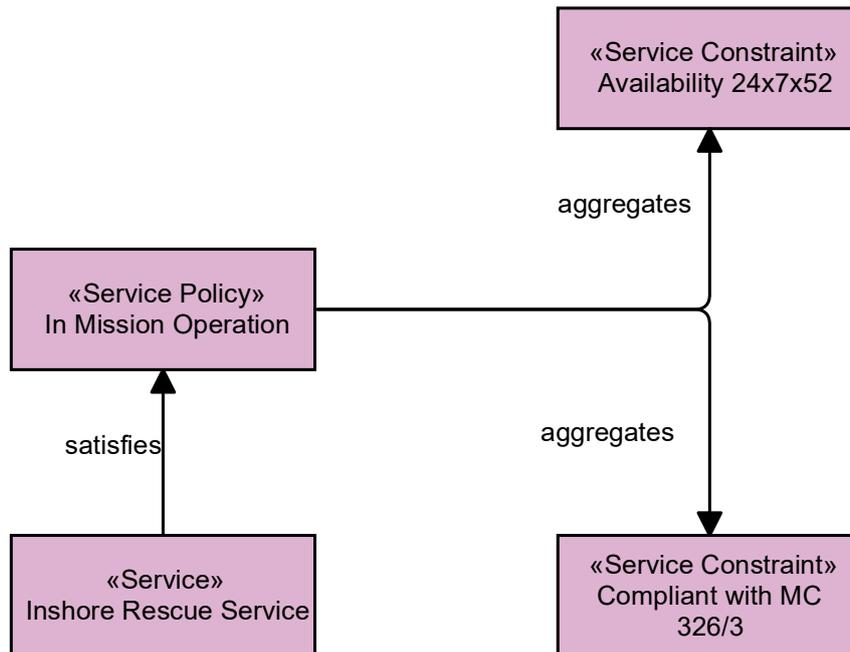
Scope
<ul style="list-style-type: none"> • Shall contain Service Constraints that are grouped by Service Policies. • May show Services that satisfy the Service Policy. • May shows Measures the Service Constraints can be evaluated by.

Representation
<ul style="list-style-type: none"> • Tabular. • Diagram. • Text Document. • Parametric diagram.

19.1 Cut-out



19.2 NAFv4 Compliant Example



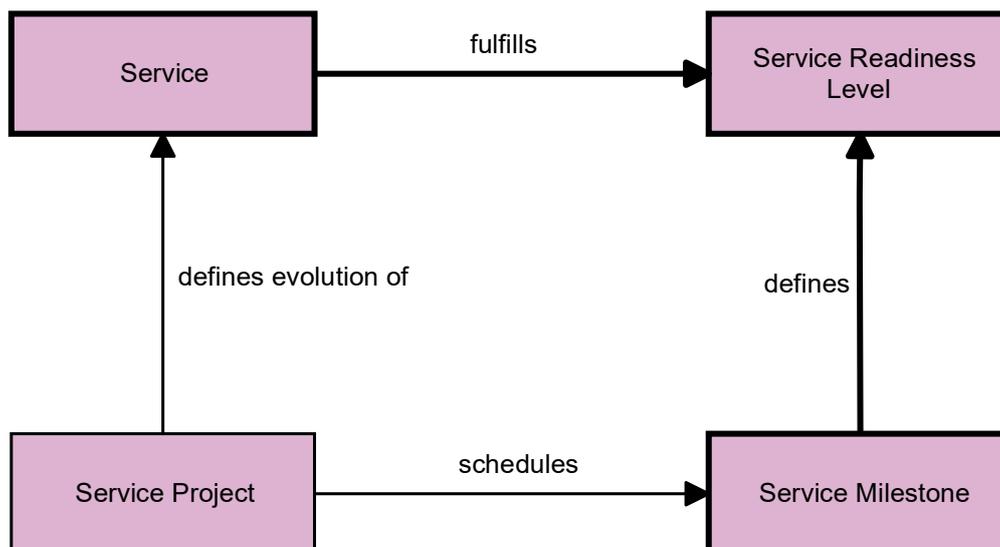
20 SR - SERVICE ROADMAP

Purpose	Usage & Concerns Addressed
The Sr Viewpoint is concerned with Service Roadmaps with Service Milestones related to the evolution of Service.	<ul style="list-style-type: none"> • Service Life Cycle Planning. • Acquisition Management. • Service phasing. • Service gap/surplus analysis. • Service orientated architecture governance.

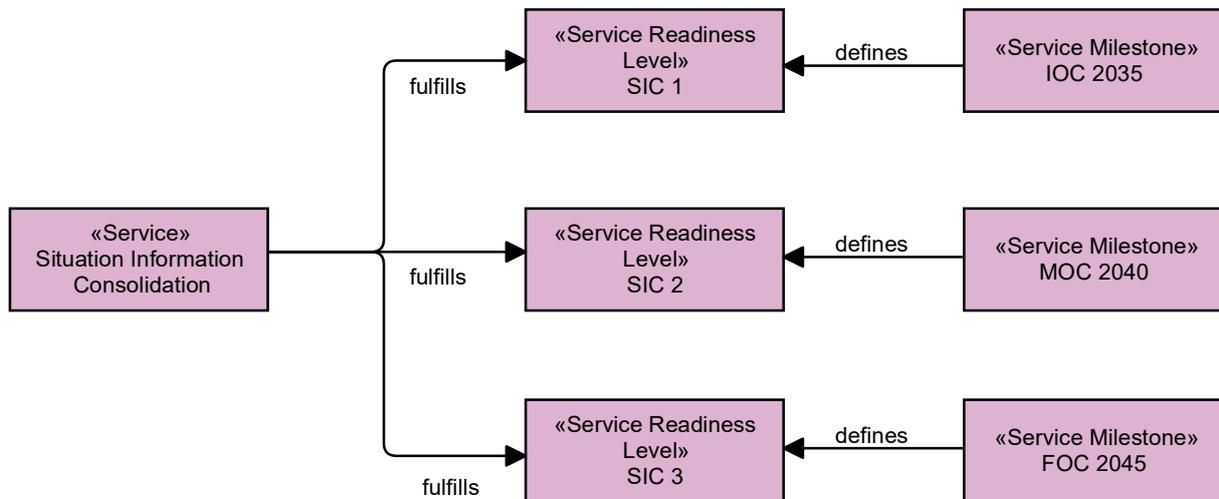
Scope
<ul style="list-style-type: none"> • Shall contain Service Milestones that define the Service Readiness Levels. • Shall contain the Services that fulfil the Service Readiness Level. • May show Service Projects and how they schedule Service Milestones and define the evolution of the Service.

Representation
<ul style="list-style-type: none"> • A time based chart in the style of a Gantt chart. • Tabular. • High-level dashboard.

20.1 Cut-out



20.2 NAFv4 Compliant Example



20.3 Notes on the Example

This example, similar to other roadmap viewpoints in NAFv4, is how to model the scenario and is important for the understanding of the evolution of the architecture over time and its impact on decisioning. However, the specific representation of the roadmap for the benefit of non-architectural stakeholder readership should be as described in the content of the 'Representation' section of this viewpoint definition.

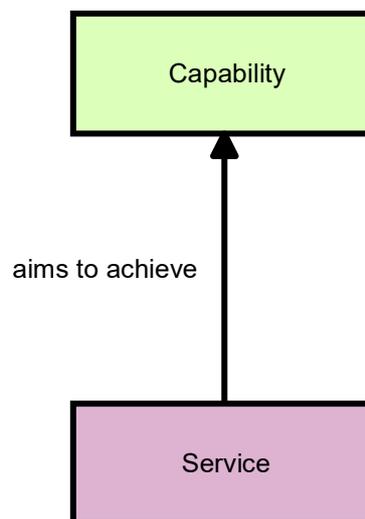
21 C1-S1 SERVICE TO CAPABILITY MAPPING

Purpose	Usage & Concerns Addressed
The C1-S1 Viewpoint is concerned with identification and description of services that enable capabilities.	<ul style="list-style-type: none"> • Mapping of capabilities to services that they are supported by. • Service Specification & Planning. • Governance.

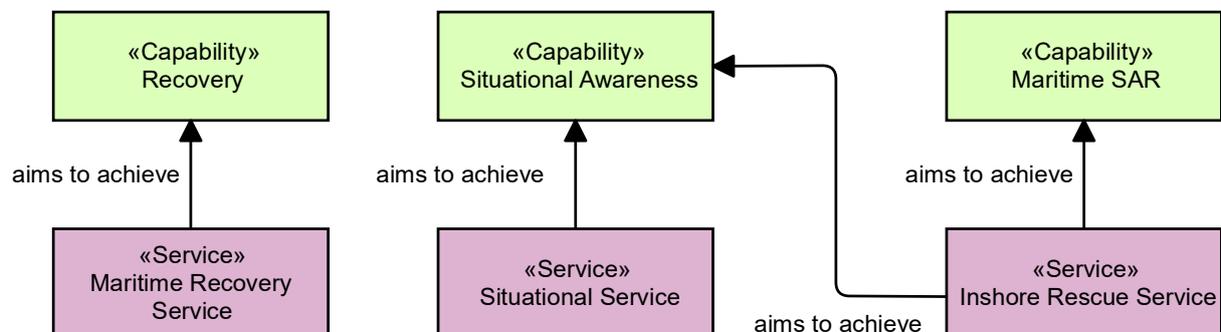
Scope
<ul style="list-style-type: none"> • Shall denote Services relevant for the architecture. • Shall denote Capabilities relevant for the architecture. • Shall link Services to Capabilities they enable or they contribute to.

Representation
<ul style="list-style-type: none"> • Matrix (with capabilities on one axis, and services on the other one). • Diagram.

21.1 Cut-out



21.2 NAFv4 Compliant Example



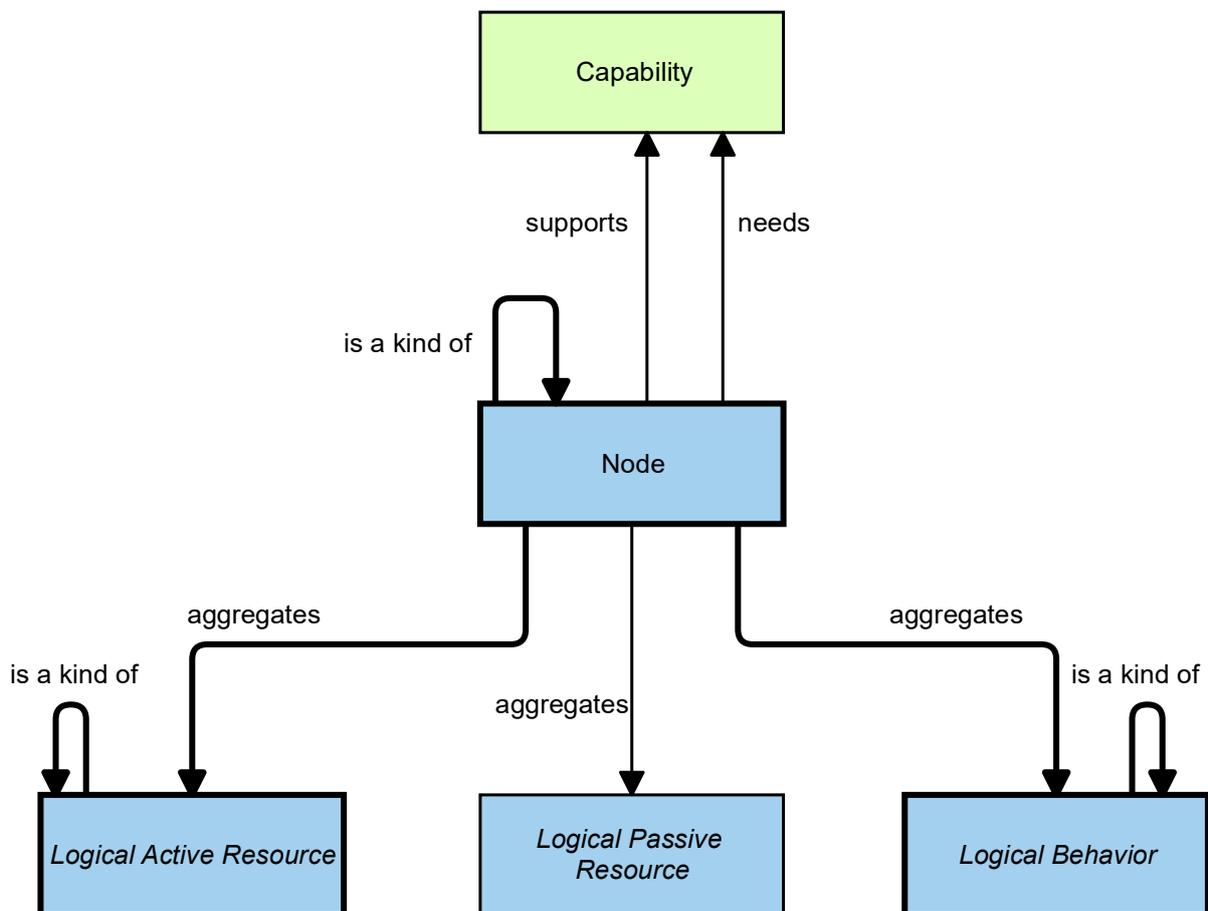
22 L1 - LOGICAL TAXONOMY

Purpose	Usage & Concerns Addressed
The L1 Viewpoint is concerned with the identification of Nodes, and their Active Resource (Logical) and Logical behaviour, and their organization into specialization taxonomies	<ul style="list-style-type: none"> • Operational Planning. • High-Level Requirements. • Initial set up of a Logical Architecture. • Defining the types of environment in which Nodes may operate.

Scope
<ul style="list-style-type: none"> • Shall contain all Nodes relevant for the entity of interest organized into specialization taxonomies. • Shall contain Logical Active Resources and Logical Behaviour that are relevant for the entity of interest, grouped by Nodes and organized into specialization taxonomies. • May show Passive Resources (Logical) that are relevant for the entity of interest, grouped by Nodes. • May trace Nodes to Capabilities they need or support.

Representation
<ul style="list-style-type: none"> • Topological (connected shapes). • Tabular.

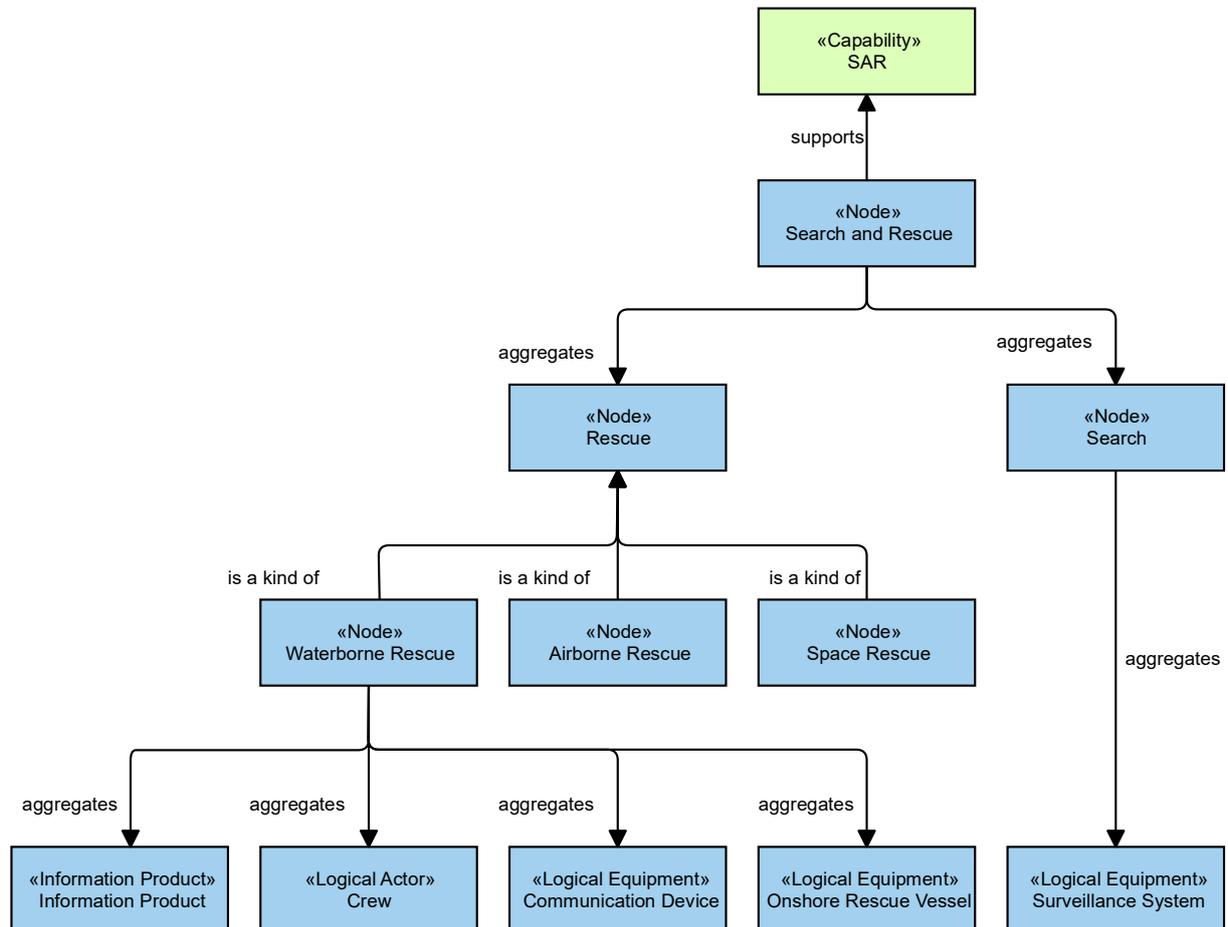
22.1 Cut-out



22.2 Notes on the Cut-out

Although Passive Logical Resources can be part of a taxonomy they are marked as optional in this viewpoint since they are more appropriately described in the L7 Information Model.

22.3 NAFv4 Compliant Example



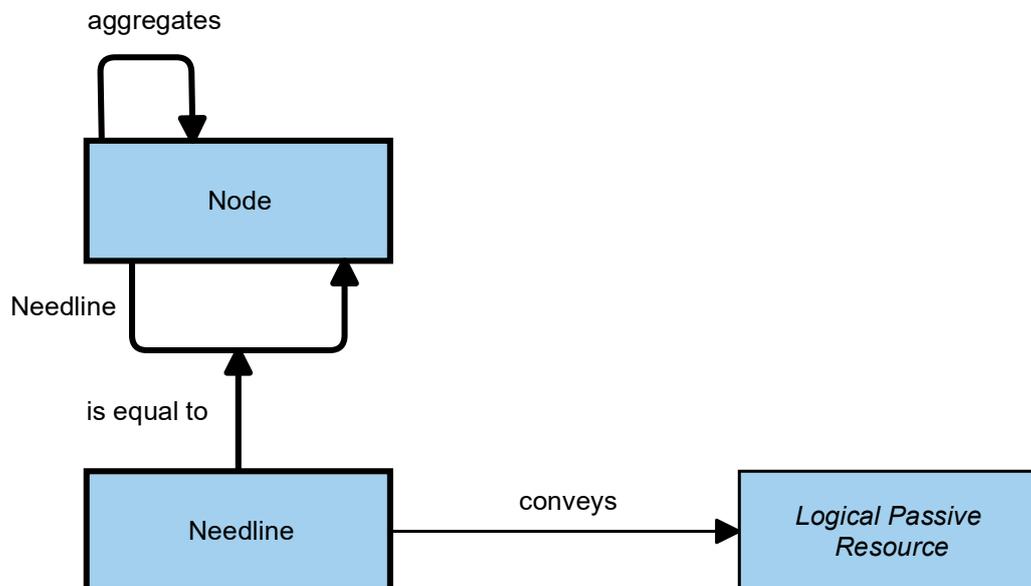
23 L2 - LOGICAL STRUCTURE

Purpose	Usage & Concerns Addressed
The L2 Viewpoint is concerned with identifying of how nodes are structured and their dependencies to other nodes through the use of Needlines for a relevant logical scenario.	<ul style="list-style-type: none"> • User Requirements. • Operational Planning. • Scenario Specification. • Definition of operational concepts. • Definition of collaboration needs. • Problem space definition. • Supply chain analysis.

Scope
<ul style="list-style-type: none"> • Shall contain Nodes relevant for the entity of interest and their structural organization. • Shall contain Needlines between Nodes. • May show Logical Passive Resources that the Needlines convey.

Representation
<ul style="list-style-type: none"> • Topological (connected shapes). • Composite structure diagram. • Rich Picture.

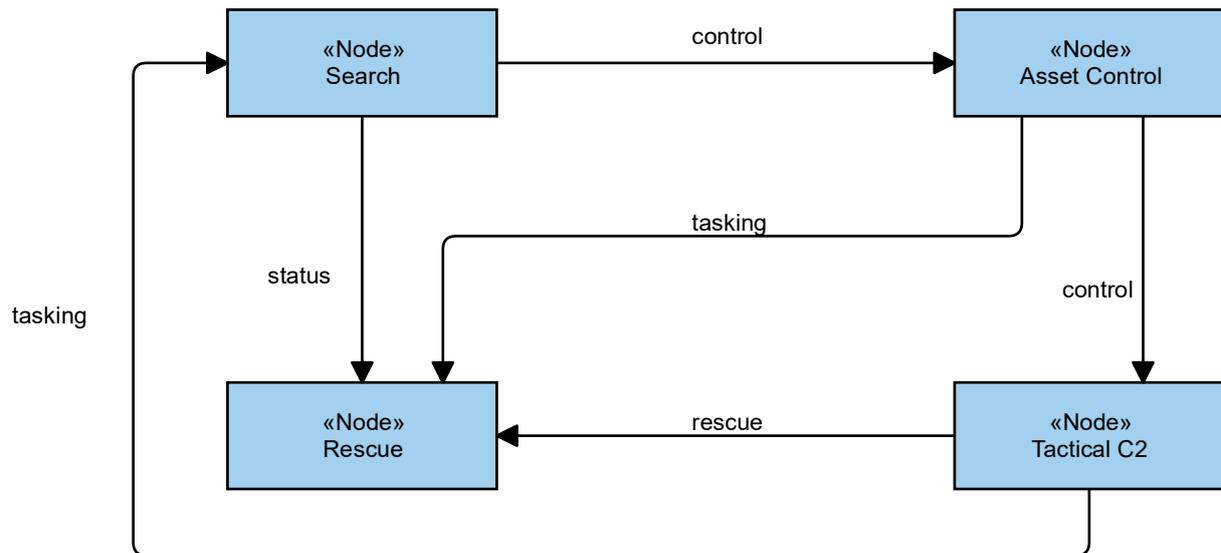
23.1 Cut-out



23.2 Notes on the Cut-out

A Needline expresses a need statement for key and/or aggregated exchanges of conveyed Items by a Node, there is no distinction into types of Node in this view, L4 elaborates on this view. The *is equal to* relationship illustrates an **objectified relation**, where the element represents the instantiated form of a connection between other elements. This construct allows the relation itself to have further relationships providing a clearer and more flexible means of expressing complex interdependencies.

23.3 NAFv4 Compliant Example

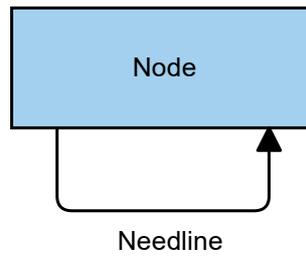


24 L2-L3 - LOGICAL CONCEPT

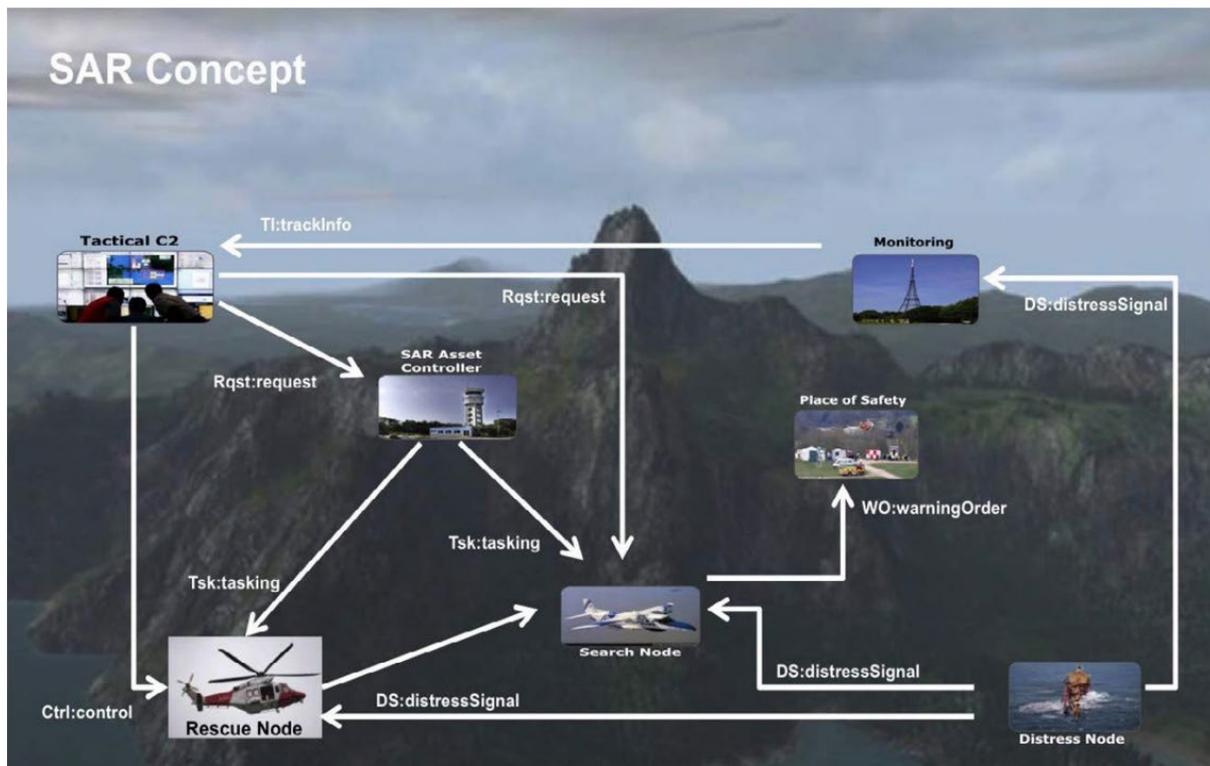
Scope

The L2-L3 is presented as a rich picture in most cases. Therefore, no information model elements are shown here. However, an L2-L3 may be constructed with a Node and it's Needlines.

24.1 Cut-out



24.2 NAFv4 Compliant Example



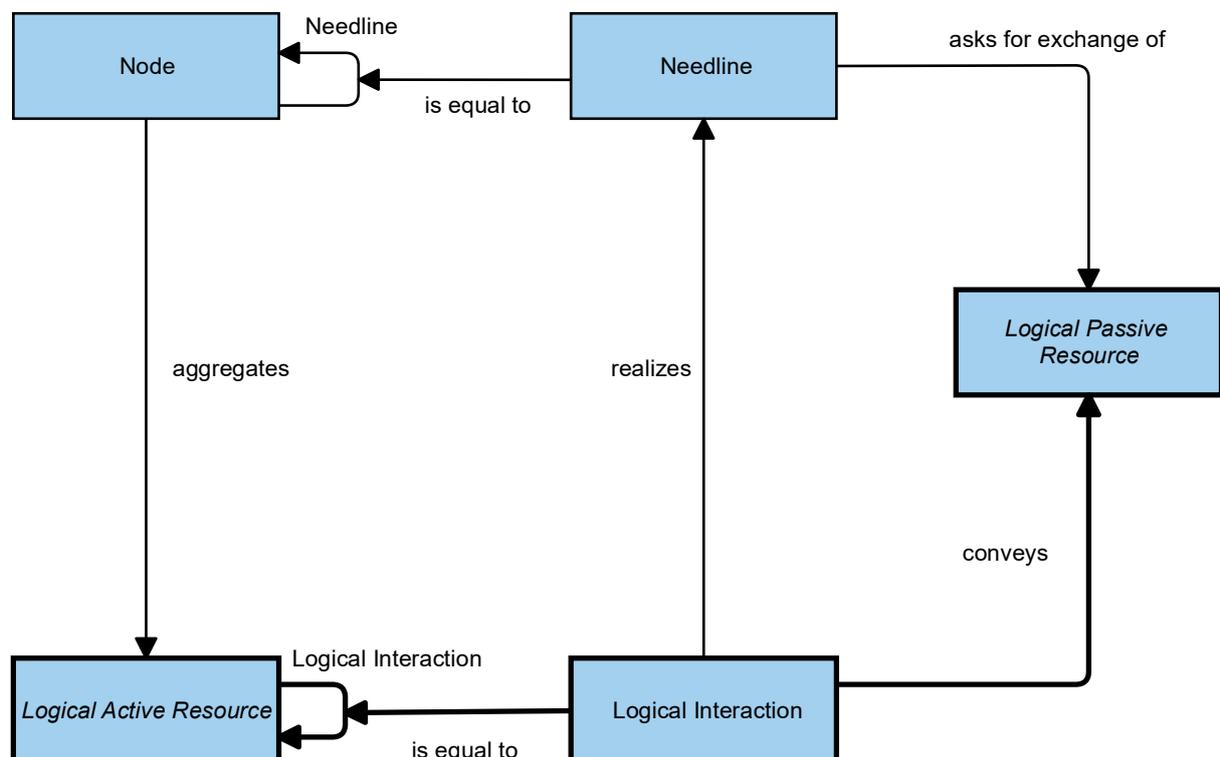
25 L3 - LOGICAL INTERACTIONS

Purpose	Usage & Concerns Addressed
The L3 Viewpoint is concerned with identifying all relevant interactions between Nodes.	<ul style="list-style-type: none"> Interoperability requirements.

Scope
<ul style="list-style-type: none"> Shall contain Logical Interactions between Logical Active Resources relevant to the entity of interest. Shall contain the Logical Passive Resources being conveyed by the Logical Interaction. May show Logical Interactions to the Needline. May show the Nodes relating to the Needlines and the Logical Active Resources they aggregate. May show the Logical Passive Resources asked for by the Needline. May show the Logical Interaction to Logical Passive Resources.

Representation
<ul style="list-style-type: none"> Tabulation. Information flow diagram. Sankey diagram.

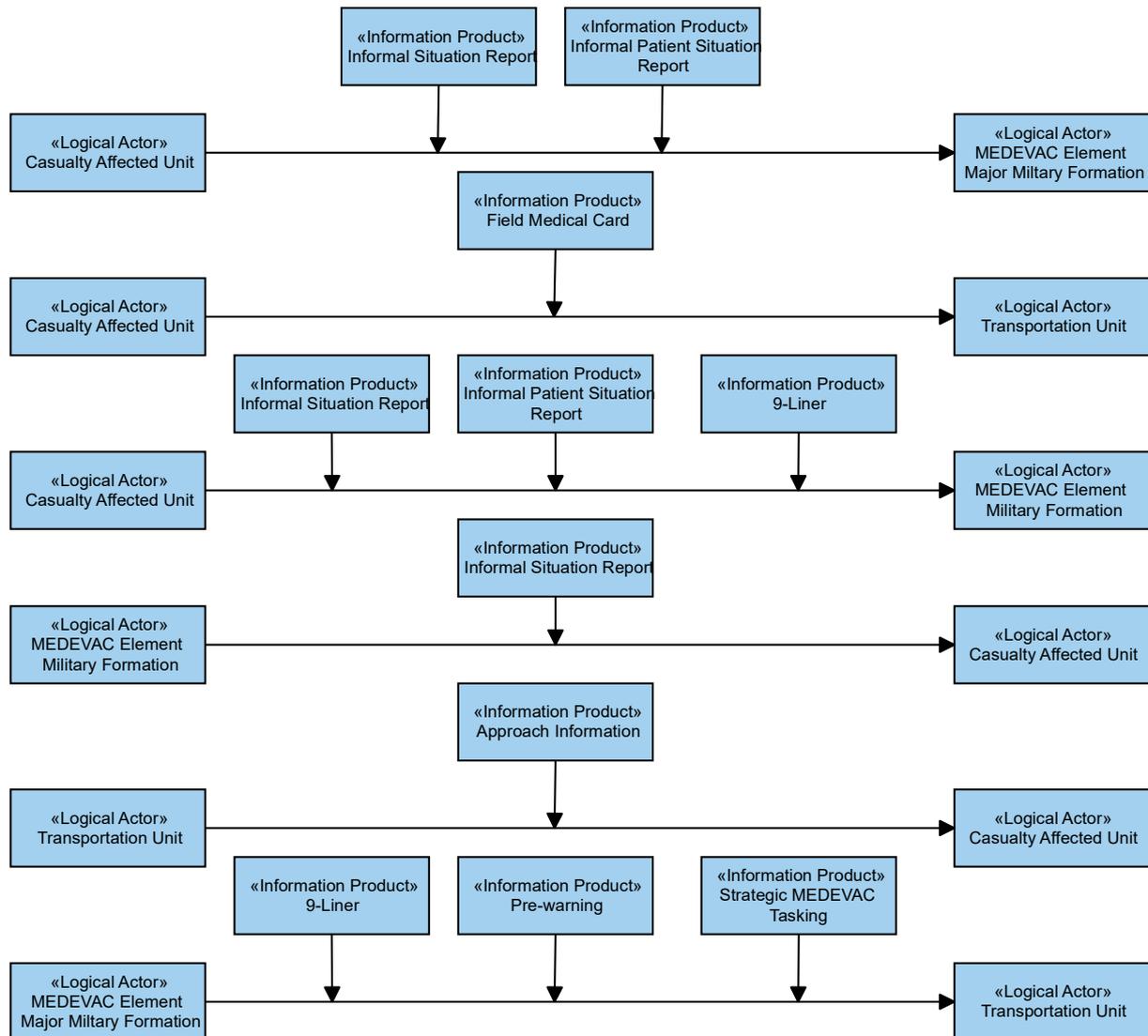
25.1 Cut-out



25.2 Notes on the Cut-out

This view elaborates on the information in L2 and L2-L3 once the information is known. The *is equal to* relationship illustrates an **objectified relation**, where the element represents the instantiated form of a connection between other elements. This construct allows the relation itself to have further relationships providing a clearer and more flexible means of expressing complex interdependencies.

25.3 NAFv4 Compliant Example



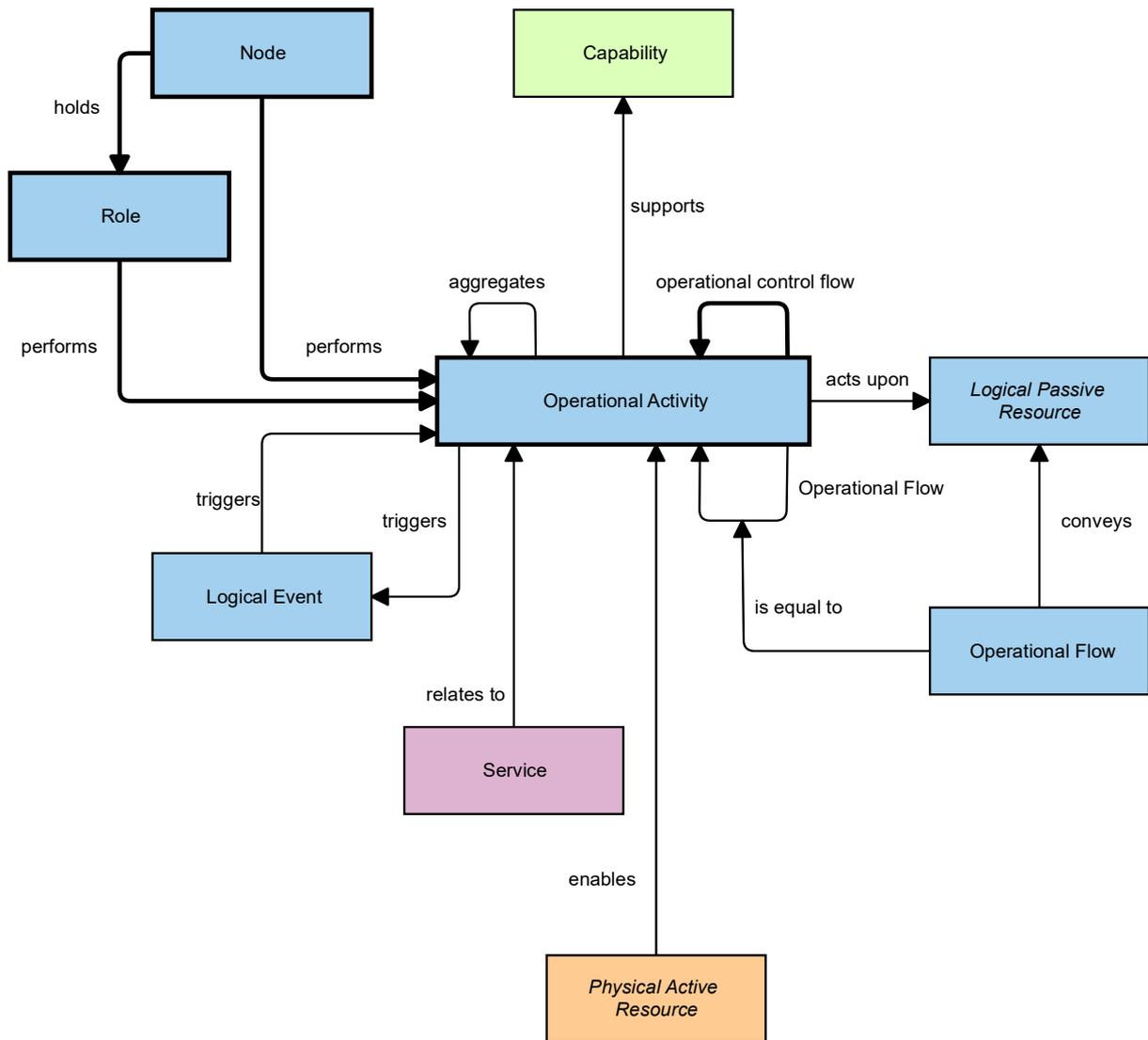
26 L4 - LOGICAL ACTIVITIES

Purpose	Usage & Concerns Addressed
<p>The L4 Viewpoint is concerned with the identification of Logical Activities, grouping and composition of these Activities and logical flows between the Activities.</p>	<ul style="list-style-type: none"> • Description of business processes and workflows. • Operational Planning. • Concept of Operations. • Service Orchestration. • Requirements capture, derivation and traceability • Logistics support analysis. • Information flow analysis. • Support task analysis to determine training needs. • Mission Threads.

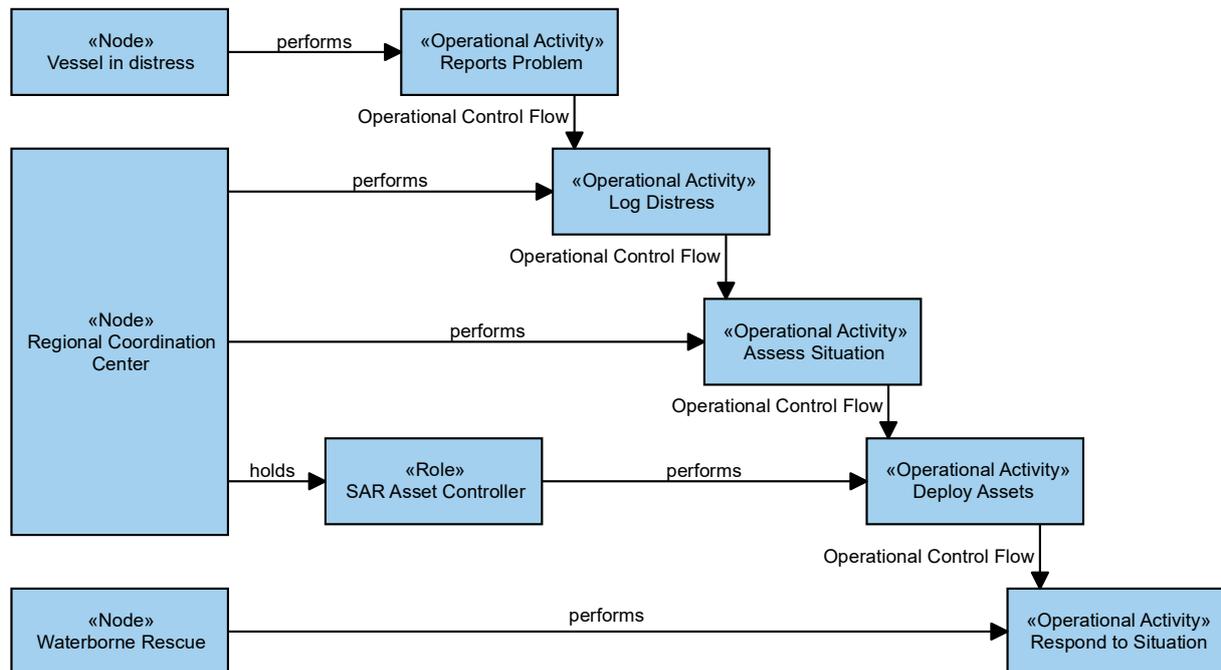
Scope
<ul style="list-style-type: none"> • Shall contain Operational Control Flows between Operational Activities. • Shall contain Nodes (or Roles that they hold) that perform Operational Activities. • May show Logical Events that trigger and can be triggered by Operational Activities. • May show Operational Flows between Operational Activities. • May show Logical Passive Resources to Operational Activities and Operational Flows. • May show the Capabilities the Operational Activities support. • May trace Physical Active Resources to the Operational Activity. • May trace Services to the Operational Activity.

Representation
<ul style="list-style-type: none"> • Hierarchy chart. • Activity diagram. • Process diagram. • Collaboration Diagram.

26.1 Cut-out



26.2 NAFv4 Compliant Example

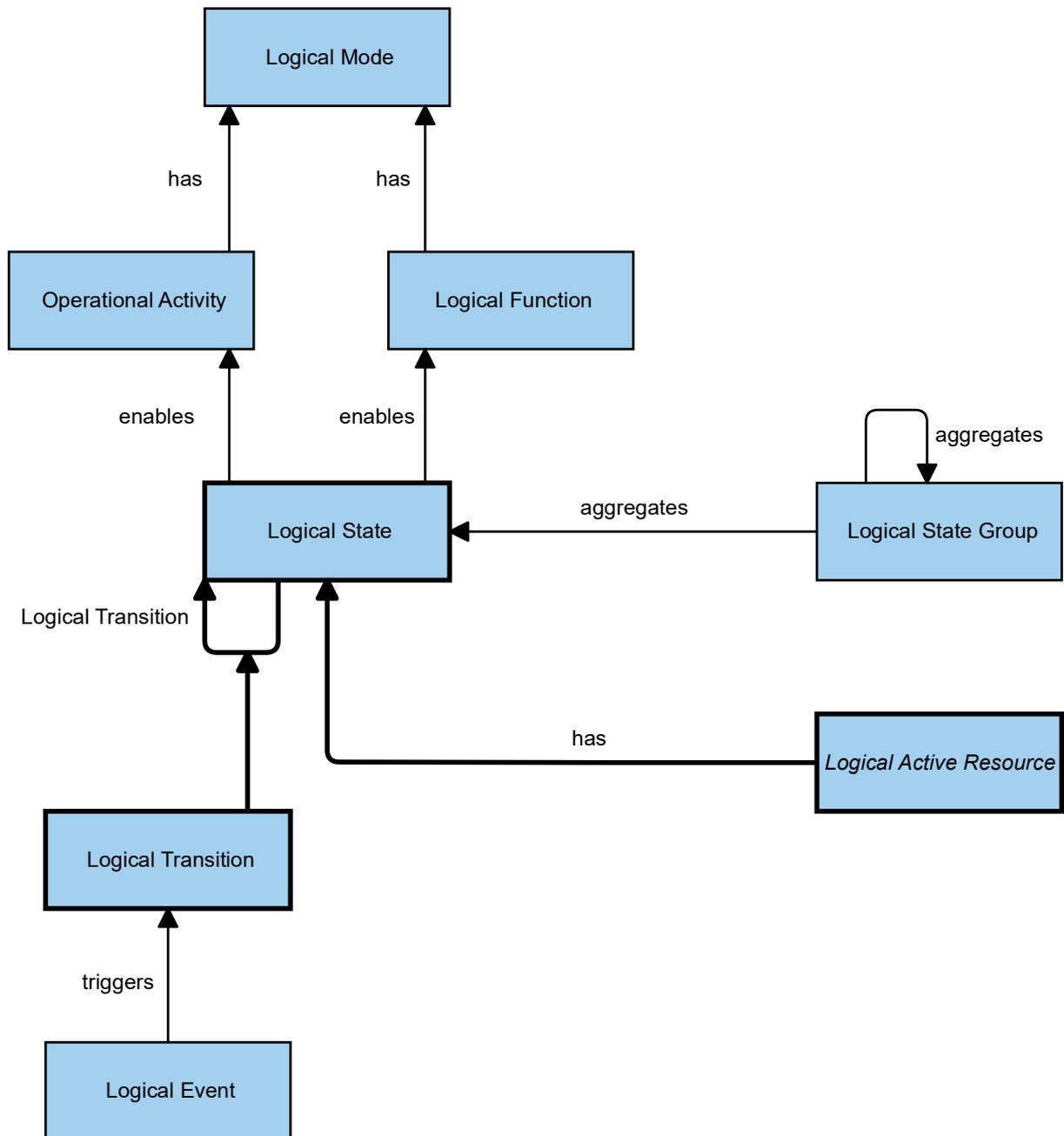


26.3 Notes on the Example

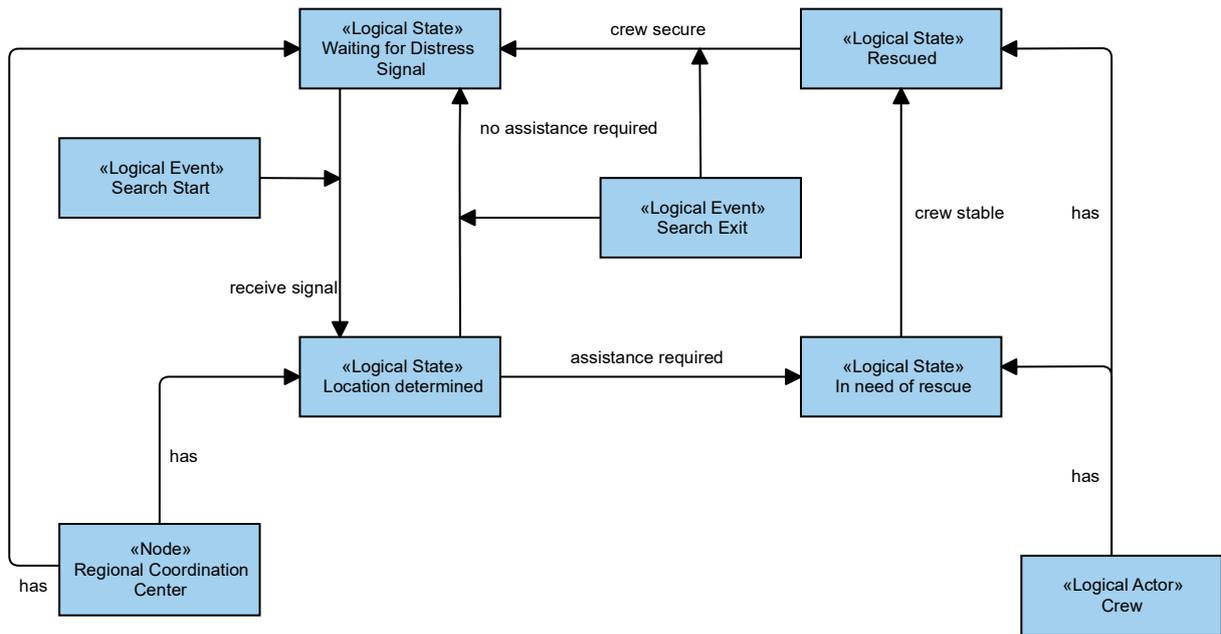
This would typically be represented as Process diagram with swim lanes.

27 L5 - LOGICAL STATES

Purpose	Usage & Concerns Addressed
The L5 Viewpoint is concerned with the identification and definition of the possible states or modes a Node may have, and the possible transitions between those States.	<ul style="list-style-type: none">• User Requirements Specification.• Analysis of business events.• Behavioural analysis.
Scope	
<ul style="list-style-type: none">• Shall contain all Logical States of Logical Active Resources relevant for the entity of interest.• Shall contain Logical Transitions between Logical States.• May show Logical events triggering the Logical Transitions.• May show Logical Modes of an Operational Activity or Logical Functions that are enabled by the Logical State.• May show Logical States Groupings of Logical States, and their groupings.	
Representation	
<ul style="list-style-type: none">• Topological (Connected Shapes).• Finite State diagram.• State transition diagram.	

27.1 Cut-out

27.2 NAFv4 Compliant Example



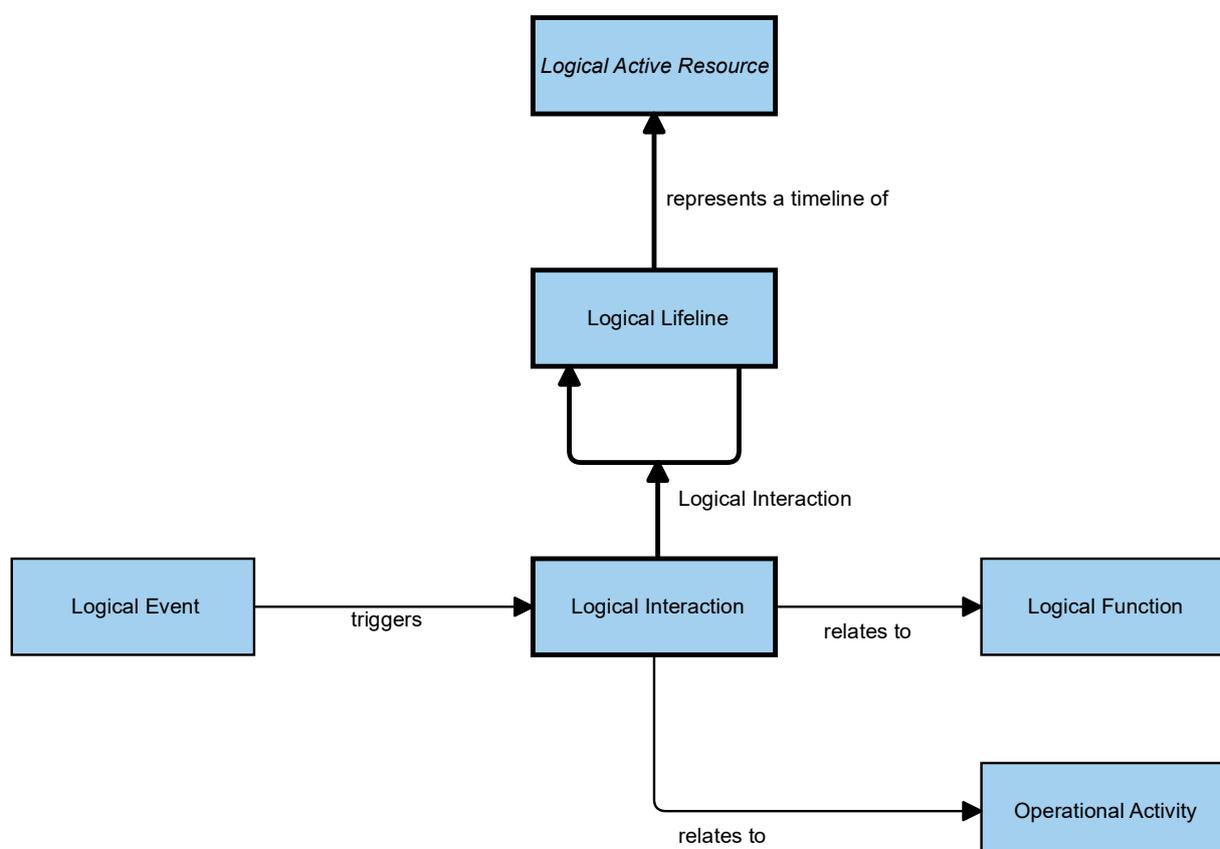
28 L6 LOGICAL SEQUENCE

Purpose	Usage & Concerns Addressed
The L6 Viewpoint is concerned with identifying the chronological sequence of activities of Active Resource (Logical) elements and/ or Logical Flows in a scenario.	<ul style="list-style-type: none"> • Operational Planning. • User Requirements Specification. • Analysis of operational events. • Sequences of interactions between nodes. • Behavioural analysis. • Service Orchestration. • Operational test scenarios.

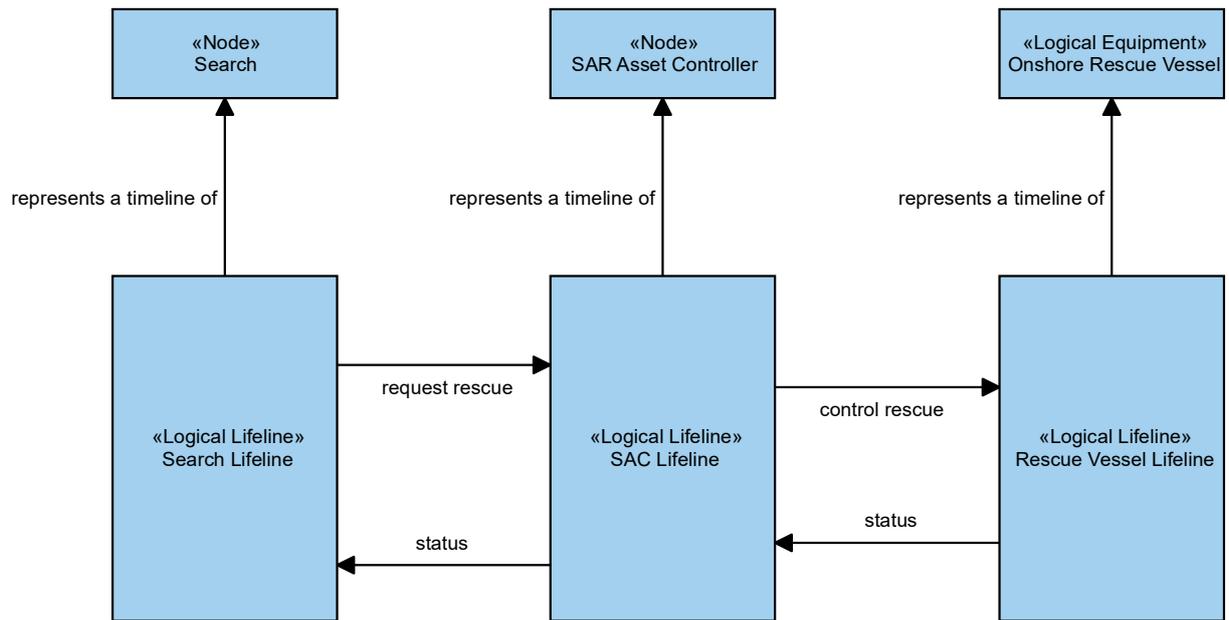
Scope
<ul style="list-style-type: none"> • Shall contain Logical Lifelines representing Logical Active Resources. • Shall contain the Logical Interactions between Logical Lifelines. • May show Logical Events that trigger Logical Interactions. • May show Operational Activities or Logical Functions related to Logical Interactions.

Representation
<ul style="list-style-type: none"> • Sequence diagram. • Event-trace diagram. • Timing diagram.

28.1 Cut-out



28.2 NAFv4 Compliant Example



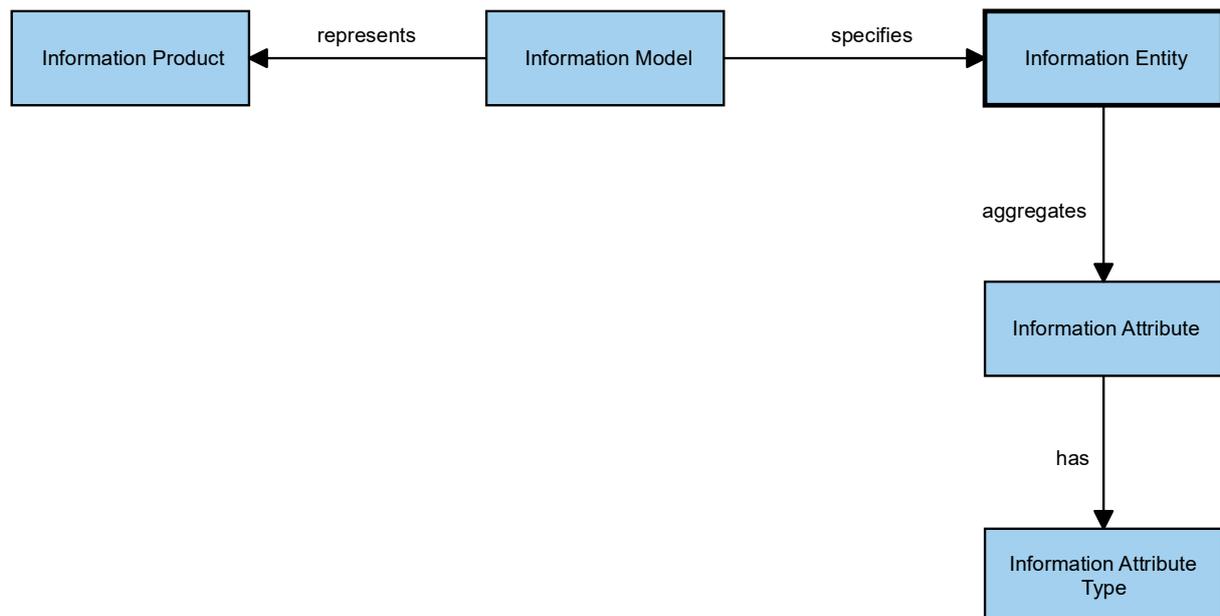
29 L7 - INFORMATION MODEL

Purpose	Usage & Concerns Addressed
The L7 Viewpoint is concerned with identifying Information Products, Information Entities and their relationships.	<ul style="list-style-type: none"> • Information Requirements. • Message Requirements. • Information Models. • Information architecture. • Information product hierarchy.

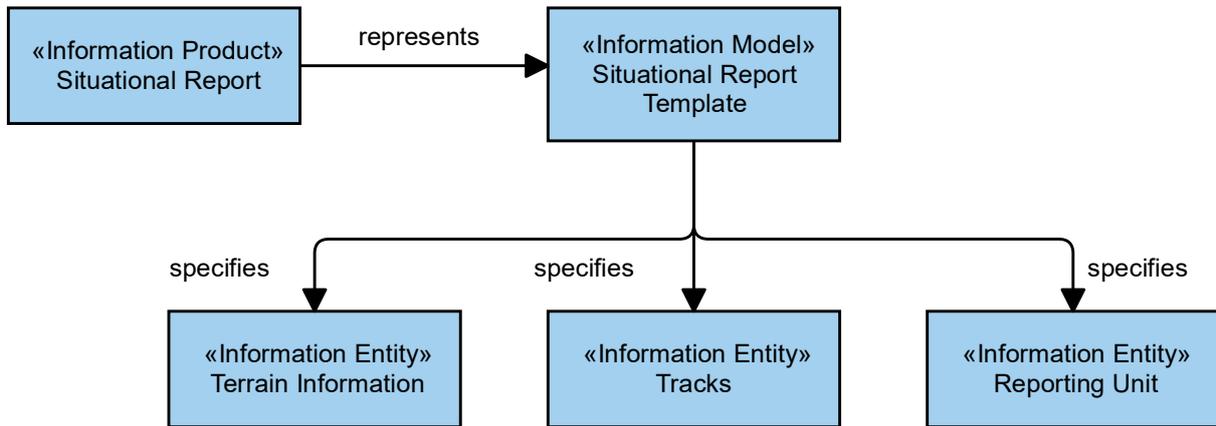
Scope
<ul style="list-style-type: none"> • Shall contain Information Entities relevant for the entity of interest. • May show an Information Model that specifies the Information Entities. • May show Information Attributes that are part of the Information Entity. • May show Information Attribute Types relating to Information Attributes. • May show Information Products that the Information Model represents.

Representation
<ul style="list-style-type: none"> • Entity-Relationship diagram. • Class diagram.

29.1 Cut-out



29.2 NAFv4 Compliant Example



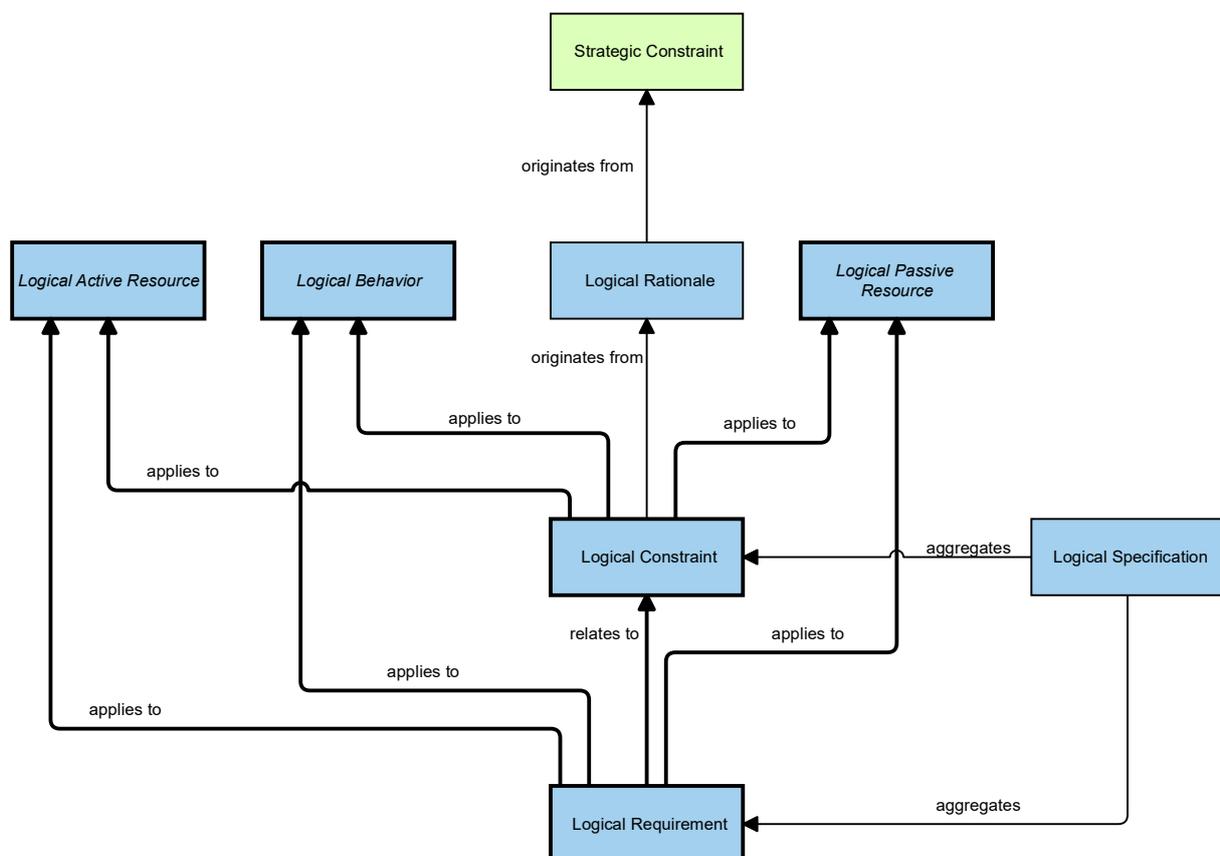
30 L8 - LOGICAL CONSTRAINTS

Purpose	Usage & Concerns Addressed
The L8 Viewpoint is concerned with the identification and description of Logical Constraints and Logical Requirements	<ul style="list-style-type: none"> • User Requirements Specification (Non-Functional). • Definition of business rules. • Identification of operational constraints.

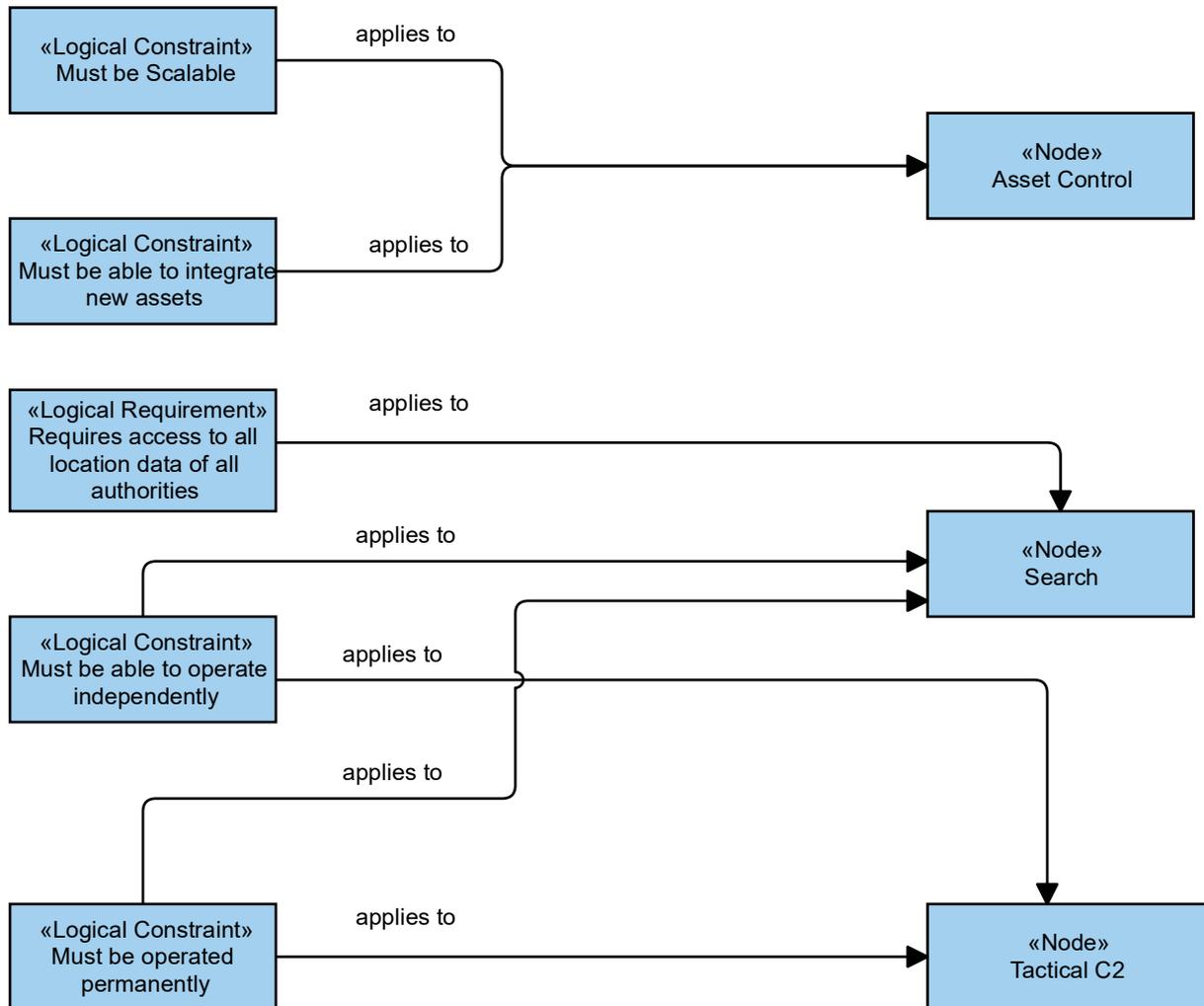
Scope
<ul style="list-style-type: none"> • Shall contain Logical Requirements that relate to Logical Constraints relevant for the entity of interest. • Shall contain Logical Active Resources, Logical Behaviour and Logical Passive Resources that these Logical Constraints or Logical Requirements apply to. • May show Logical Specifications that group Logical Constraints and Logical Requirements. • May show Logical Rationale relating to Logical Constraints. • May trace Logical Rationale to Strategic Constraints.

Representation
<ul style="list-style-type: none"> • Structured Text. • Business rules diagram. • Matrix Table. • Parametric diagram.

30.1 Cut-out



30.2 NAFv4 Compliant Example



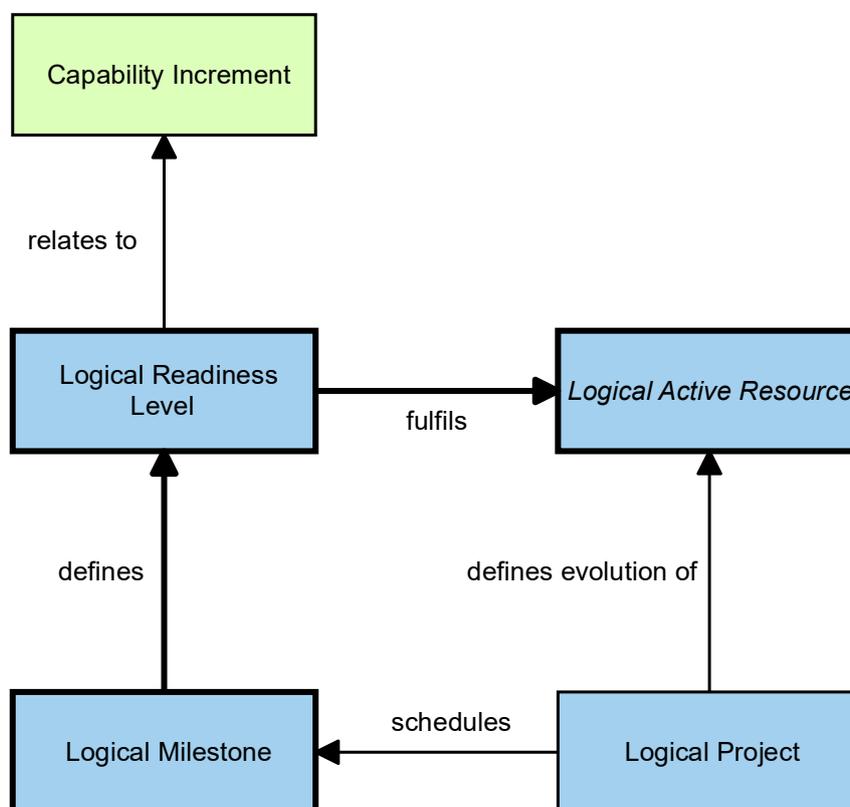
31 LR - LOGICAL ROADMAP

Purpose	Usage & Concerns Addressed
The Lr Viewpoint is concerned with Identifying Logical Roadmaps with Logical Milestones related to the evolution of Resource Building blocks	<ul style="list-style-type: none"> • Acquisition Planning. • Portfolio / Programme Management. • Project Performance Reporting. • Project management and control (including delivery timescales). • Project dependencies and the identification of associated risk. • Through Life Management Planning.

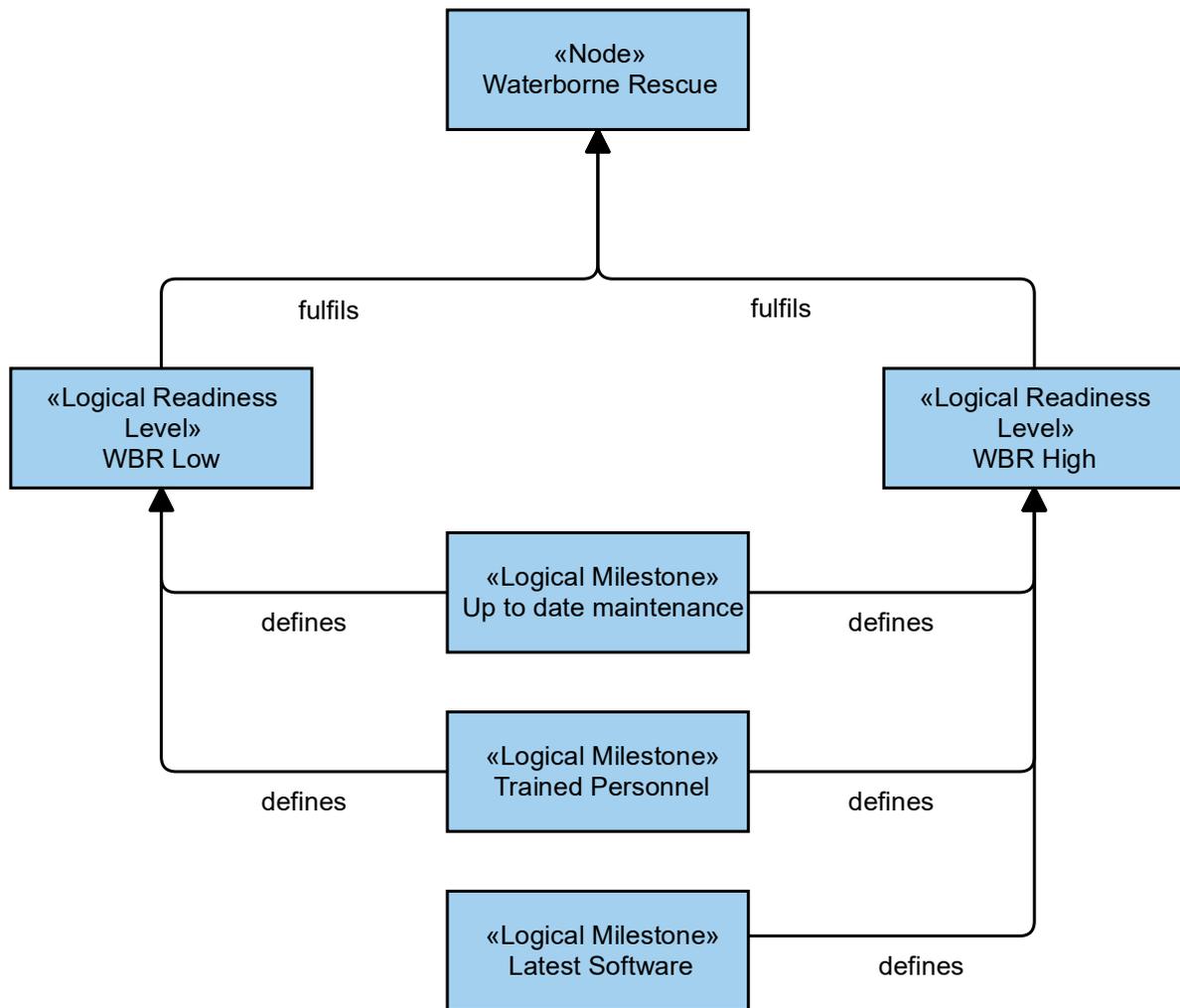
Scope
<ul style="list-style-type: none"> • Shall contain Logical Milestones that define the Logical Readiness Levels. • Shall contain the Logical Active Resources that the Logical Readiness Level fulfils. • May trace Logical Readiness Levels to Capability Increments. • May show Logical Projects and how they schedule Logical Milestones and define the evolution of the Logical Active Resource.

Representation
<ul style="list-style-type: none"> • Timeline View. • Augmented chart in style of a Gantt Chart. • Portfolio Dashboard.

31.1 Cut-out



31.2 NAFv4 Compliant Example



31.3 Notes on the Example

This example, similar to other roadmap viewpoints in NAFv4, is how to model the scenario and is important for the understanding of the evolution of the architecture over time and its impact on decisioning. However, the specific representation of the roadmap for the benefit of non-architectural stakeholder readership should be as described in the content of the 'Representation' section of this viewpoint definition.

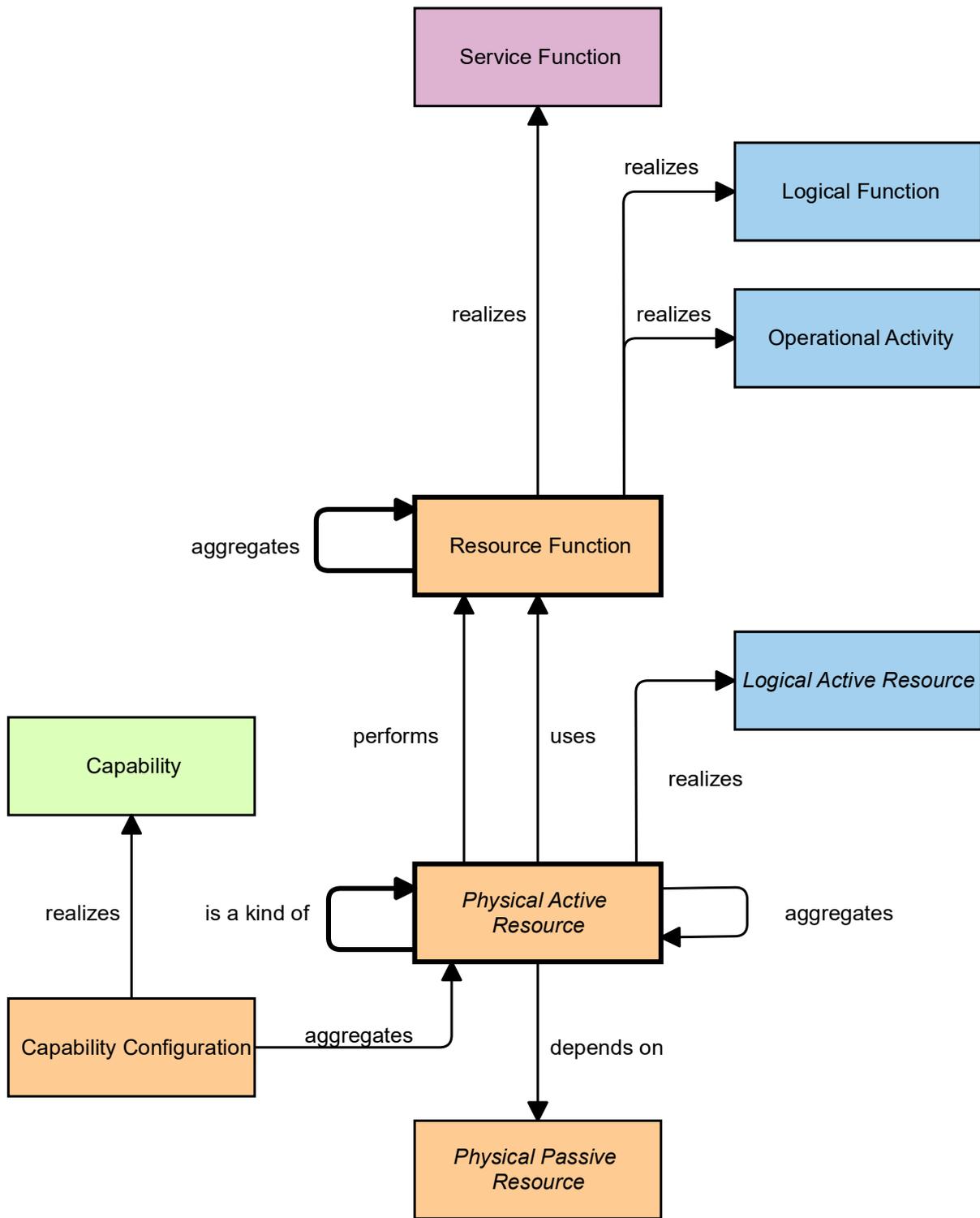
32 P1 - RESOURCE TAXONOMY

Purpose	Usage & Concerns Addressed
<p>The P1 Viewpoint is concerned with the Identification of Active Resources (Physical) and Physical Behaviour and their mapping to Conceptual, Service and Logical building blocks and their organization into taxonomies.</p>	<ul style="list-style-type: none"> • Capability Delivery. • Service Implementation. • Identifying Resource Taxonomies. • Identification of applicable protocols. • Forecasting technology readiness against time. • HR trends analysis. • Recruitment planning. • Planning technology insertion. • Input to options analysis.

Scope
<ul style="list-style-type: none"> • Shall contain all Physical Active Resources organized into specialization taxonomies and Resource Functions, and their groupings, relevant for the entity of interest. • May show linkages between Physical Active Resources and Resource Functions. • May show Physical Active Resources organized into Capability Configurations. • May show Physical Passive Resources that Physical Active Resources depend on. • May trace Capabilities to Capability Configurations. • May trace Physical Active Resources to Logical Active Resources. • May trace Resource Functions to Service Functions and Logical Behaviour.

Representation
<ul style="list-style-type: none"> • Tabular. • Mapping (matrix). • Composite Structure Diagram. • Block diagram. • Class Diagram

32.1 Cut-out



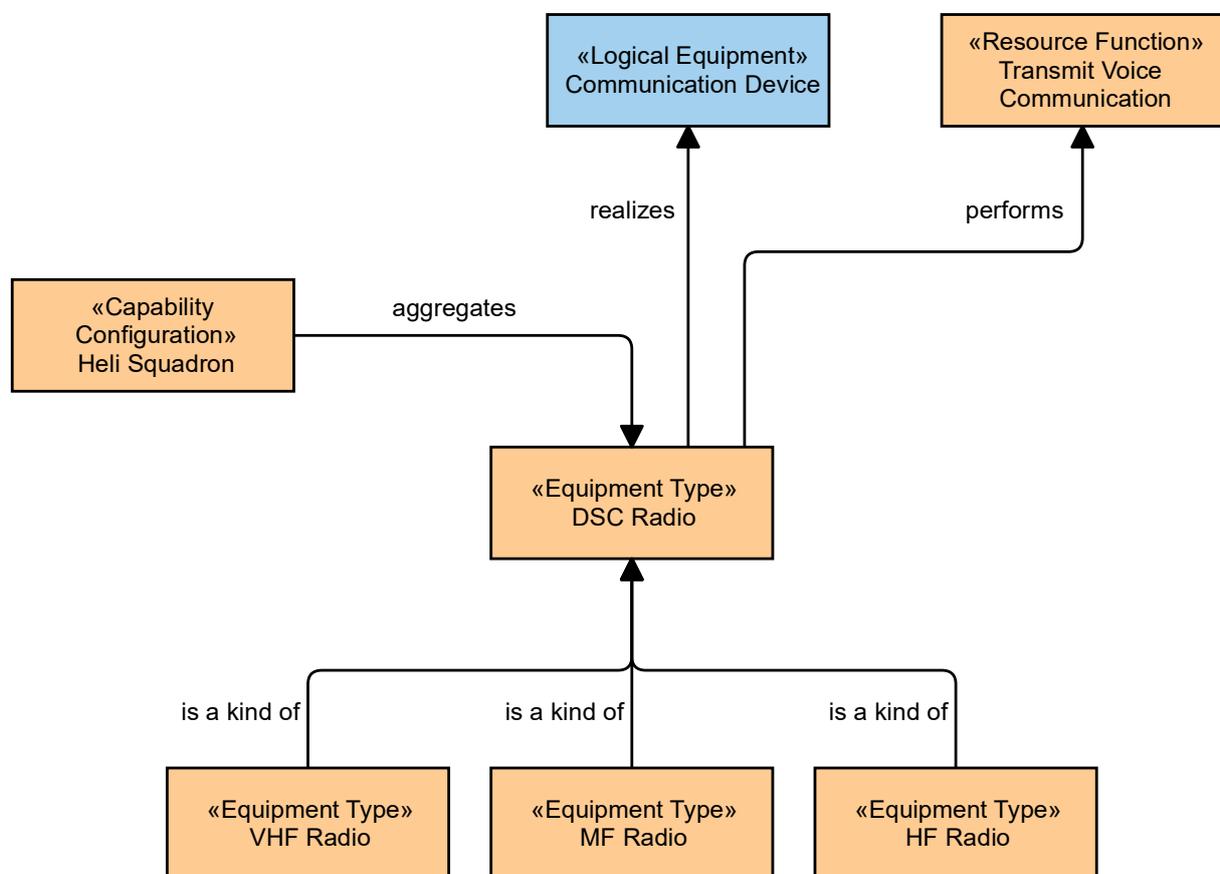
32.2 Notes on the Cut-out

Systems, Networks, Ports and Physical Assets from NAFv3 can be represented as subtypes of Physical Active Resources in the equipment and technology dimension. Physical Assets from NAFv3 can also be represented as subtypes of Physical Passive Resources in the material dimension.

The focus of the view is taxonomies of resources, therefore the relations between mandatory elements are optional.

During the development of architectures views may be revisited multiple times. For example, in this view specific Capability Configurations may be unknown initially, but may emerge over time and can therefore be included.

32.3 NAFv4 Compliant Example



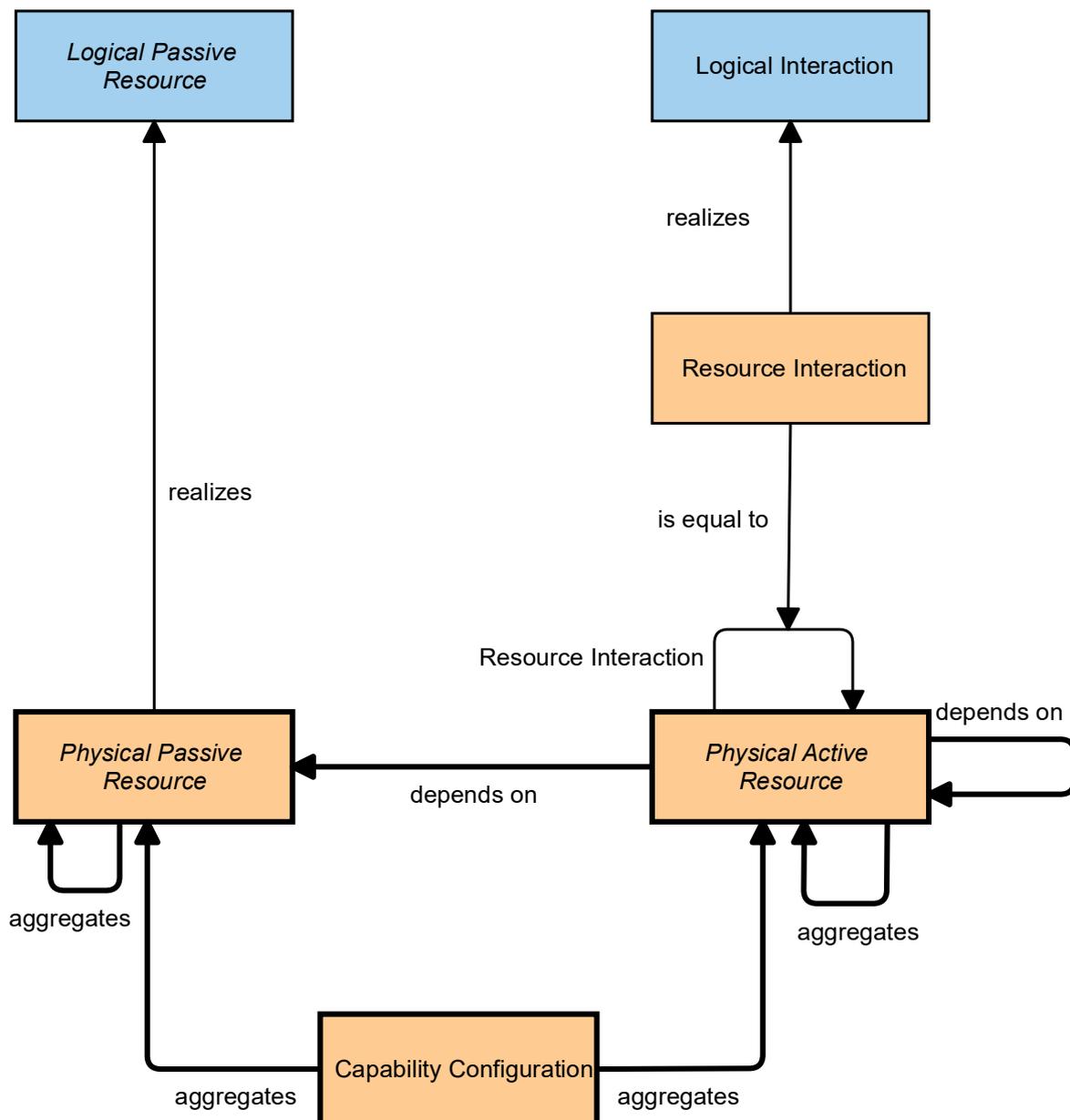
33 P2 - RESOURCE STRUCTURE

Purpose	Usage & Concerns Addressed
<p>The P2 viewpoint is concerned with the identification of how resources are structured and their dependencies to other resources,</p>	<ul style="list-style-type: none"> • Physical Architecture. • Systems Engineering / Design. • Organizational Design. • Systems Integration. • System Requirements Specification. • Definition of system concepts. • Definition of system options. • Human – System interactions. • Interface requirements capture. • Capability integration planning. • System integration management. • Operational planning (capability configuration definition).

Scope
<ul style="list-style-type: none"> • Shall contain Physical Active Resources and Physical Passive Resources organized into Capability Configurations. • Shall contain the structural organization of Physical Active Resources and Physical Passive Resources. • Shall contain dependencies between Physical Active Resources and Physical Passive Resources. • May show Resource interactions between Physical Active Resources. • May trace Resource Interactions to Logical Interactions defined in L3. • May trace Physical Passive Resources to Logical Passive Resources.

Representation
<ul style="list-style-type: none"> • Composite structure diagram. • Block diagram. • Internal Block diagram. • Physical Block diagram. • Class diagram.

33.1 Cut-out



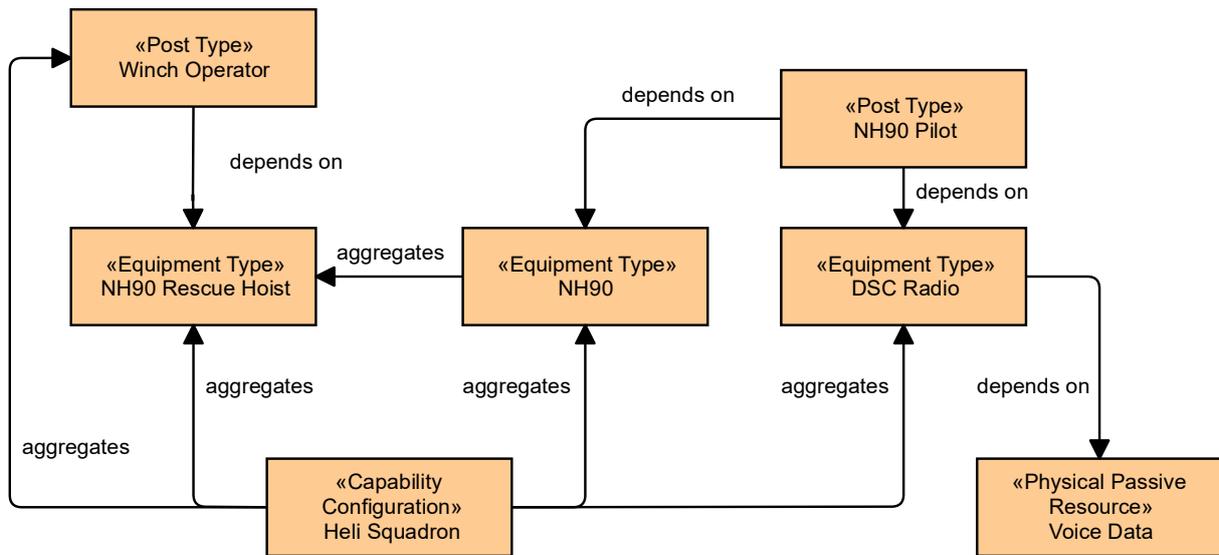
33.2 Notes on the Cut-out

The mapping between Resource Interactions and Needlines is expressed transitively through the Logical Interactions.

If there are multiple solutions satisfying the operational needs specified in the L2, they can be depicted in different P2 views.

The *is equal to* relationship illustrates an **objectified relation**, where the element represents the instantiated form of a connection between other elements. This construct allows the relation itself to have further relationships providing a clearer and more flexible means of expressing complex interdependencies.

33.3 NAFv4 Compliant Example



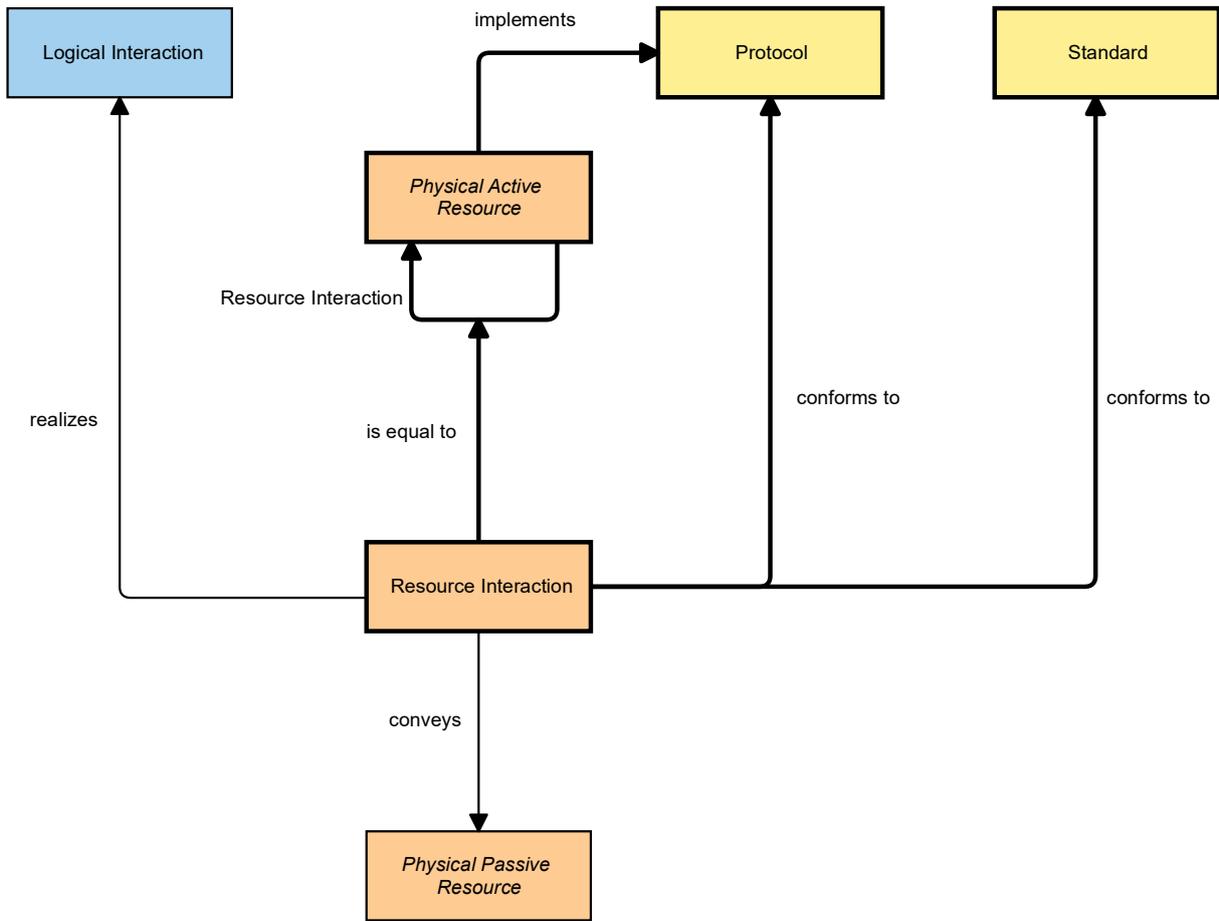
34 P3 - RESOURCE INTERACTIONS

Purpose	Usage & Concerns Addressed
<p>The P3 Viewpoint is concerned with identifying all relevant interactions between active resources.</p>	<ul style="list-style-type: none"> • Interface Specification. • Systems Engineering. • System Requirements. • Identification of interactions. • Identification of applicable protocols. • Description of system communication paths. • Bandwidth and capacity analysis. • Detailed definition of data flows. • Detailed definition of resource flows.

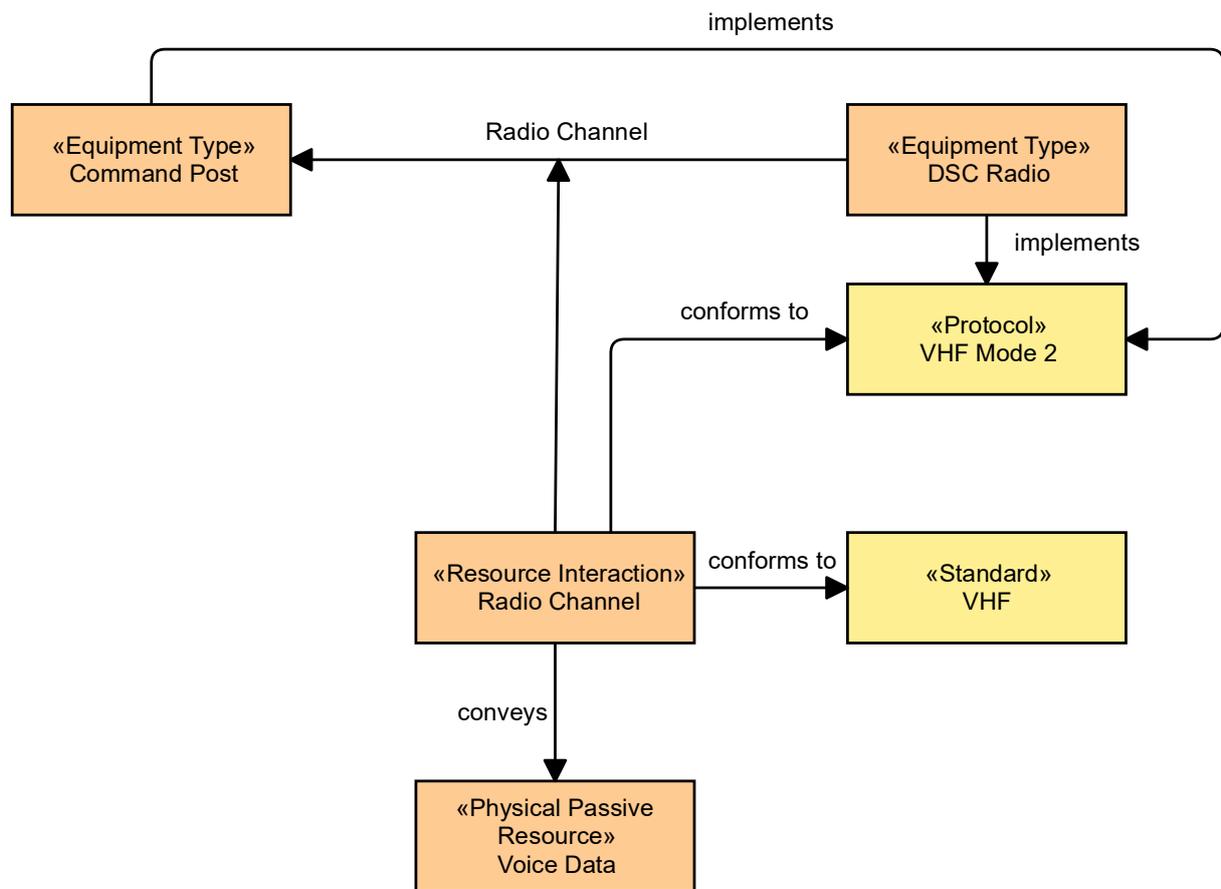
Scope
<ul style="list-style-type: none"> • Shall contain the Resource Interactions between the Physical Active Resources. • Shall contain the Standards and Protocols the Resource Interaction conforms to. • Shall contain the Protocols that the Physical Active Resource implements. • May trace Resource Interactions to Logical Interactions. • May show Physical Passive Resources that the Resource Interaction conveys.

Representation
<ul style="list-style-type: none"> • Block diagram. • Internal Block diagram. • Physical Block diagram. • Class diagram. • Tabular.

34.1 Cut-out



34.2 NAFv4 Compliant Example



35 P4 - RESOURCE FUNCTIONS

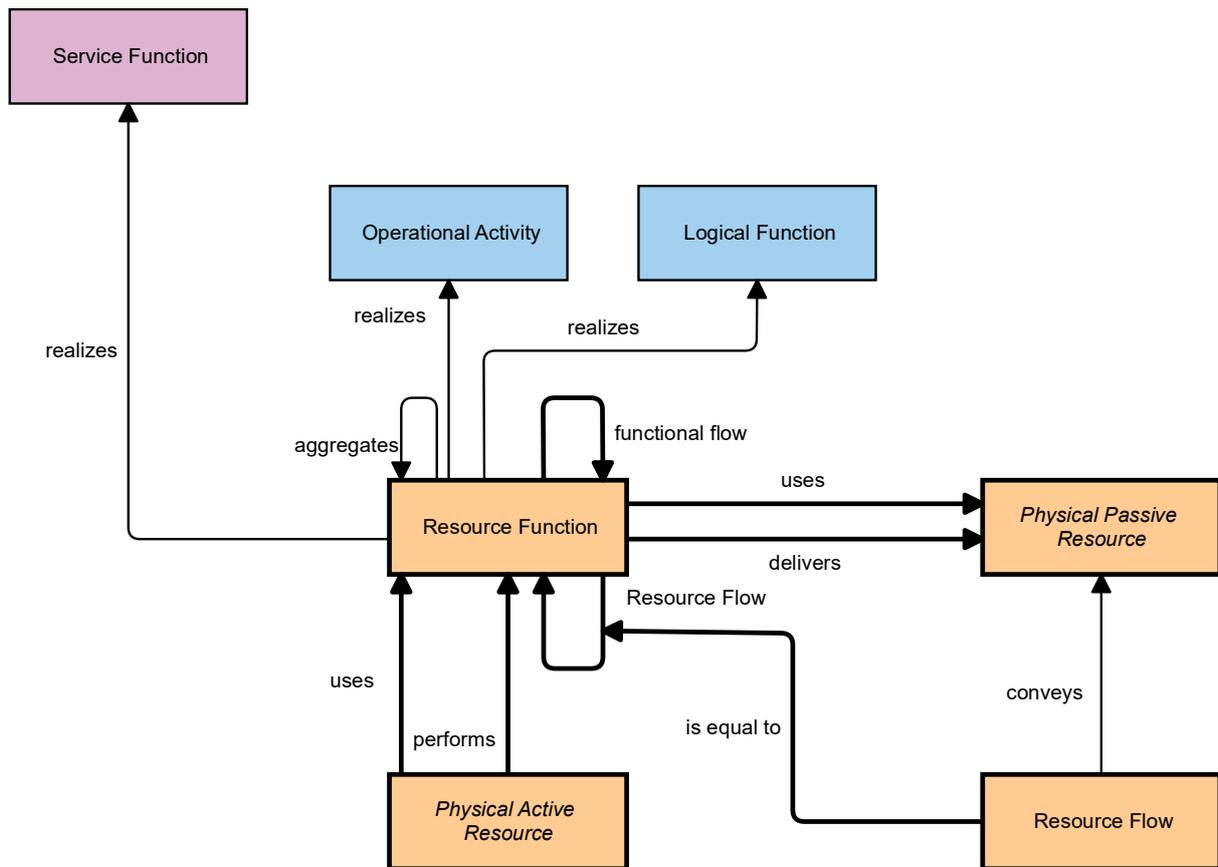
Purpose	Usage & Concerns Addressed
<p>The P4 Viewpoint is concerning the Identification of the Resource Functions carried out by Resources, and Resource Flows between Resources</p>	<ul style="list-style-type: none"> • Capability-Based Acquisition. • Business Process Modelling. • Workflow Modelling. • Human-Machine Interaction Specifications. • Description of task workflow. • Identification of functional system requirements. • Functional decomposition of systems. • Relate human and system functions.

Scope

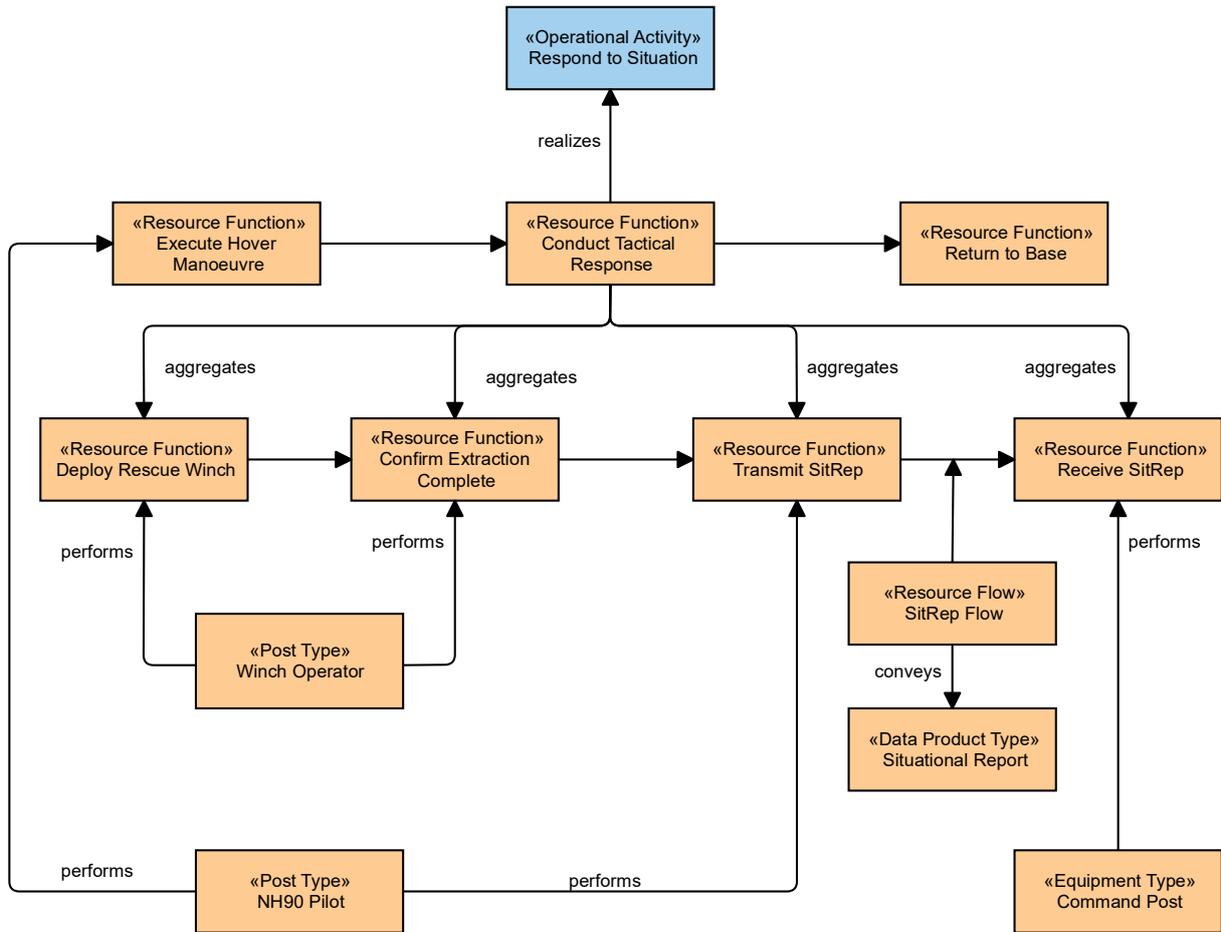
- Shall contain the Physical Active Resources that use or perform a Resource Function.
- Shall contain the Physical Passive Resources that the Resource Function uses or delivers.
- Shall contain Functional Flows and Resource Flows between Resource Functions.
- May show the Physical Passive Resources that the Resource Flow conveys.
- May trace Resource Functions to Logical Behaviour or Service Functions.

Representation

- Topological (connected shapes).
- Activity diagram.
- Collaboration diagram.
- Process diagram.
- Functional Breakdown (decomposition).

35.1 Cut-out

35.2 NAFv4 Compliant Example



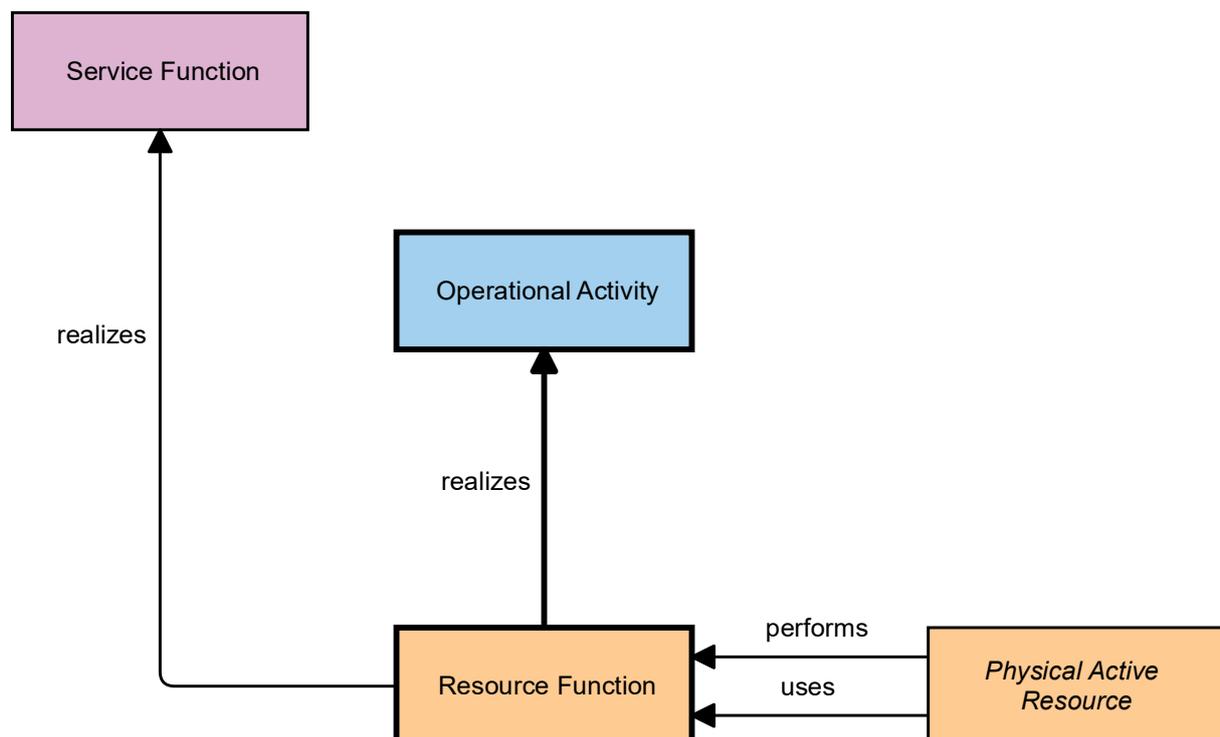
36 L4-P4 - ACTIVITY TO FUNCTION MAPPING

Purpose	Usage & Concerns Addressed
The L4-P4 Viewpoint is concerned with addressing the linkage between Resource Functions and Operational Activities specified in L4 as well as the linkage between Resource Functions and Service Functions specified in the S4.	<ul style="list-style-type: none"> • Requirements Definition. • Process Mapping. • Tracing functional system requirements to user requirements. • Tracing solution options to requirements. • Identification of overlaps.

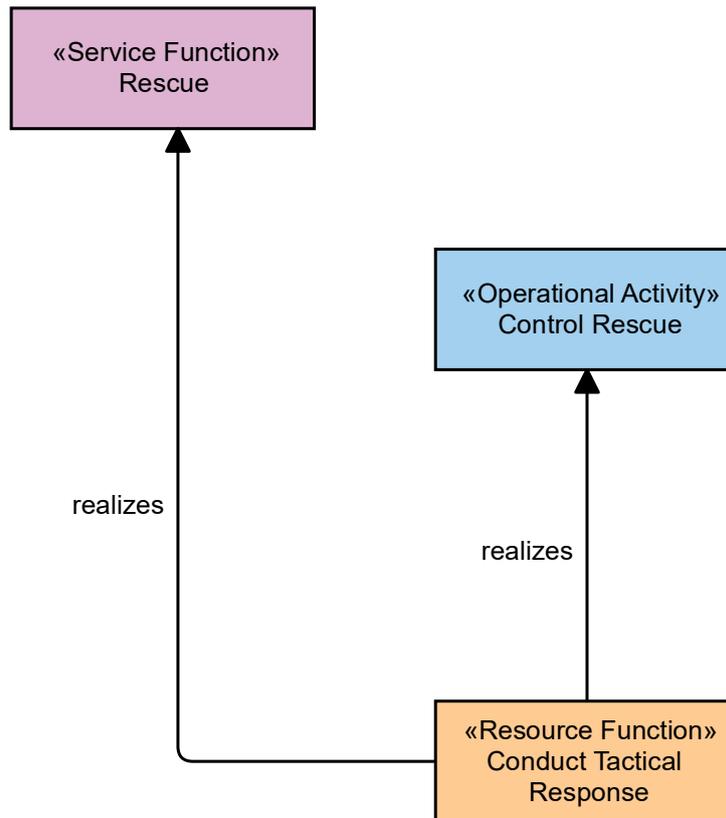
Scope
<ul style="list-style-type: none"> • Shall contain the traceability between Resource Functions and Operational Activities. • May trace Resource Functions to Service Functions defined in the S4. • May show Physical Active Resources that use or perform the Resource Functions.

Representation
<ul style="list-style-type: none"> • Tabular. • Matrix. • Diagram.

36.1 Cut-out



36.2 NAFv4 Compliant Example



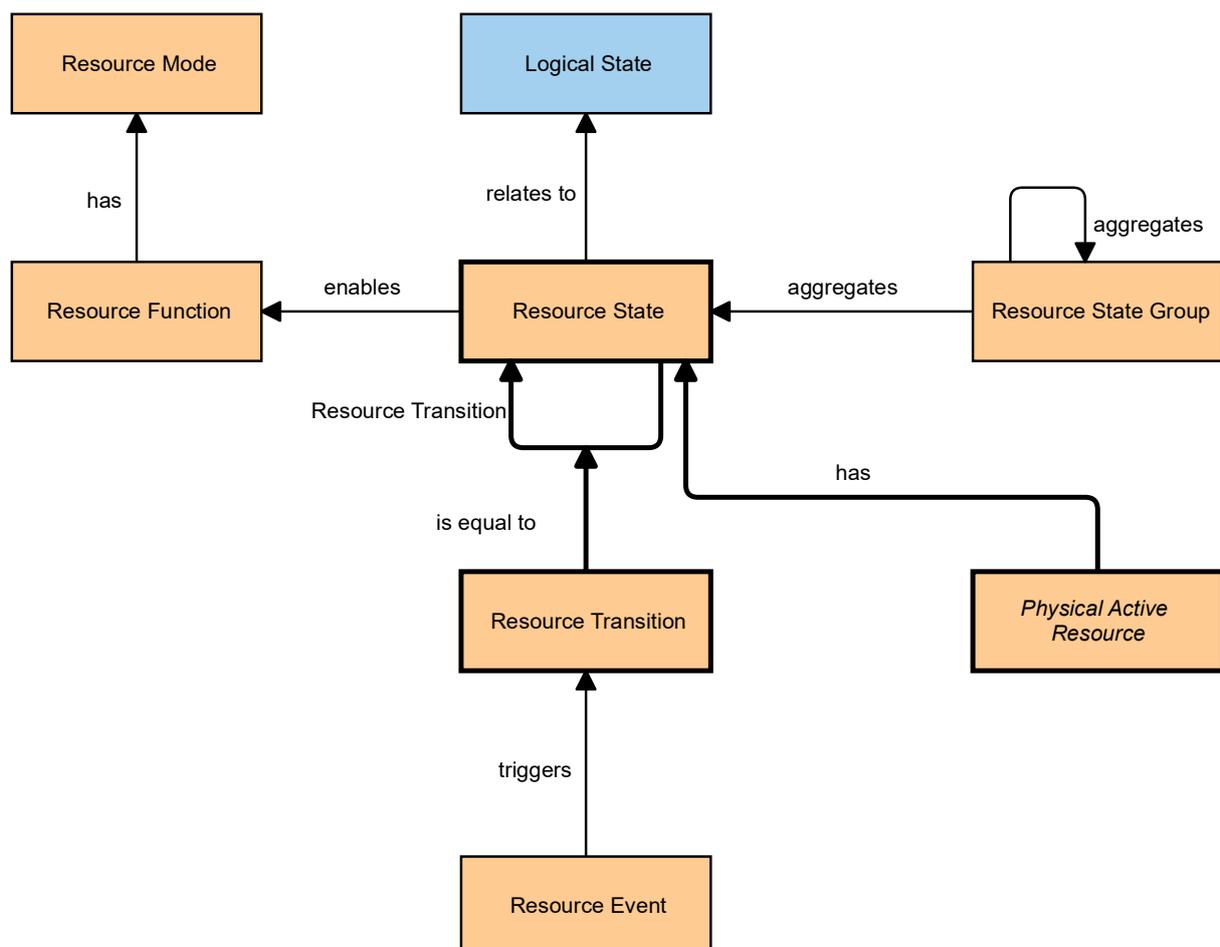
37 P5 - RESOURCE STATES

Purpose	Usage & Concerns Addressed
The P5 Viewpoint is concerned with Resource Types changing State in response to events and other stimuli.	<ul style="list-style-type: none"> • Systems Engineering. • Safety Cases. • Definition of states, events and state transitions (behavioural modeling).

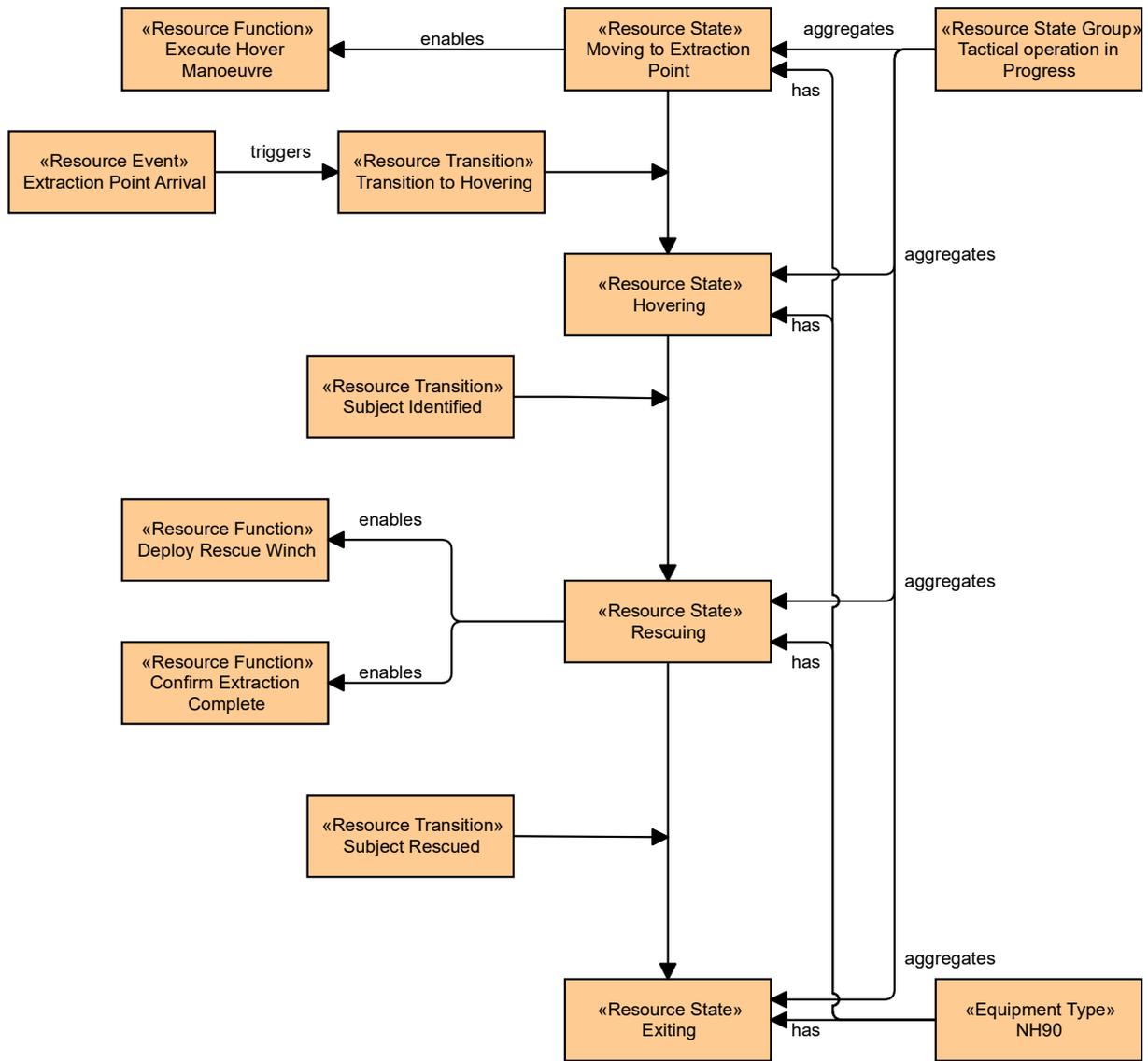
Scope
<ul style="list-style-type: none"> • Shall contain all Resource States of Physical Active Resources relevant for the entity of interest. • Shall contain Resource Transitions between Resource States. • May show Resource Events triggering Resource Transitions. • May show Resource Modes of a Resource Functions that are enabled by the Resource State. • May show Resource States Groupings of Resource States, and their groupings. • May trace Resource States to Logical States.

Representation
<ul style="list-style-type: none"> • State Transition diagram. • Finite State diagram.

37.1 Cut-out



37.2 NAFv4 Compliant Example



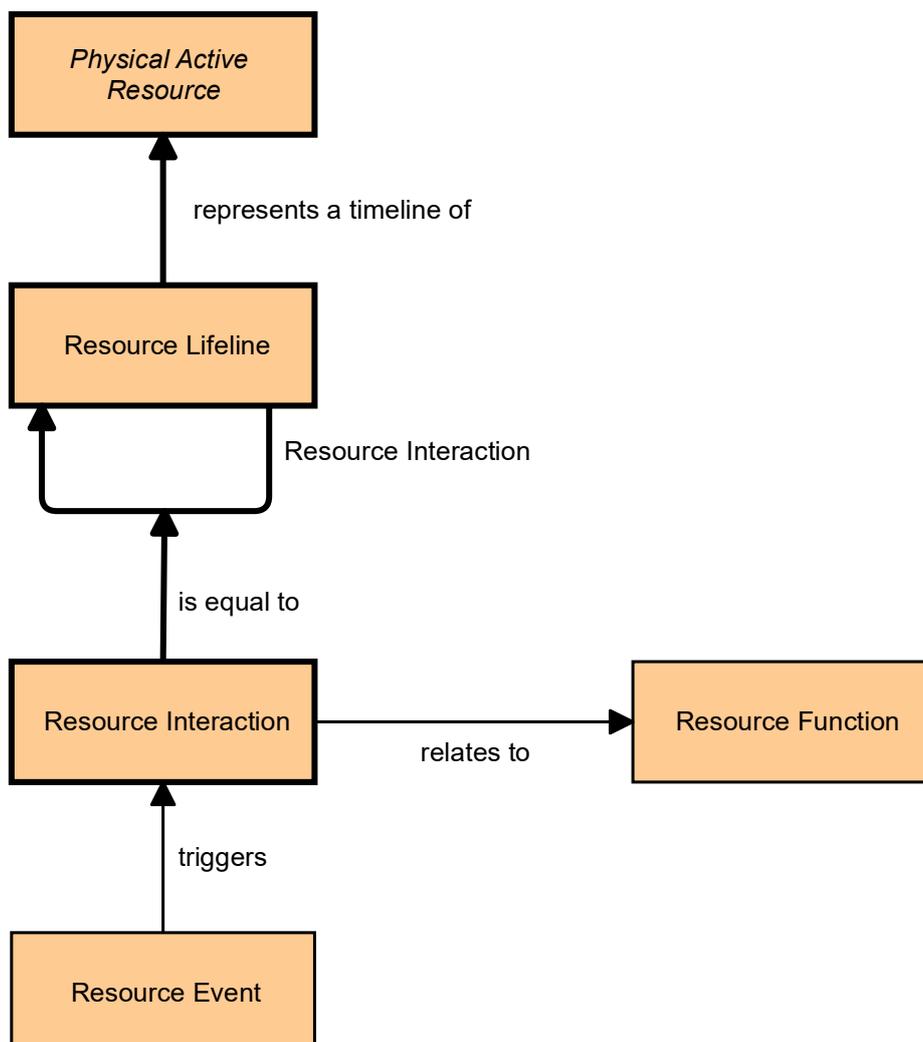
38 P6 RESOURCE SEQUENCE

Purpose	Usage & Concerns Addressed
The P6 Viewpoint is concerned with identifying the chronological sequence of the Interactions between active Resources in a scenario.	<ul style="list-style-type: none"> • Message Handling. • Complex System Behaviours. • Security Modelling. • Analysis of resource events impacting operation. • Behavioural analysis.

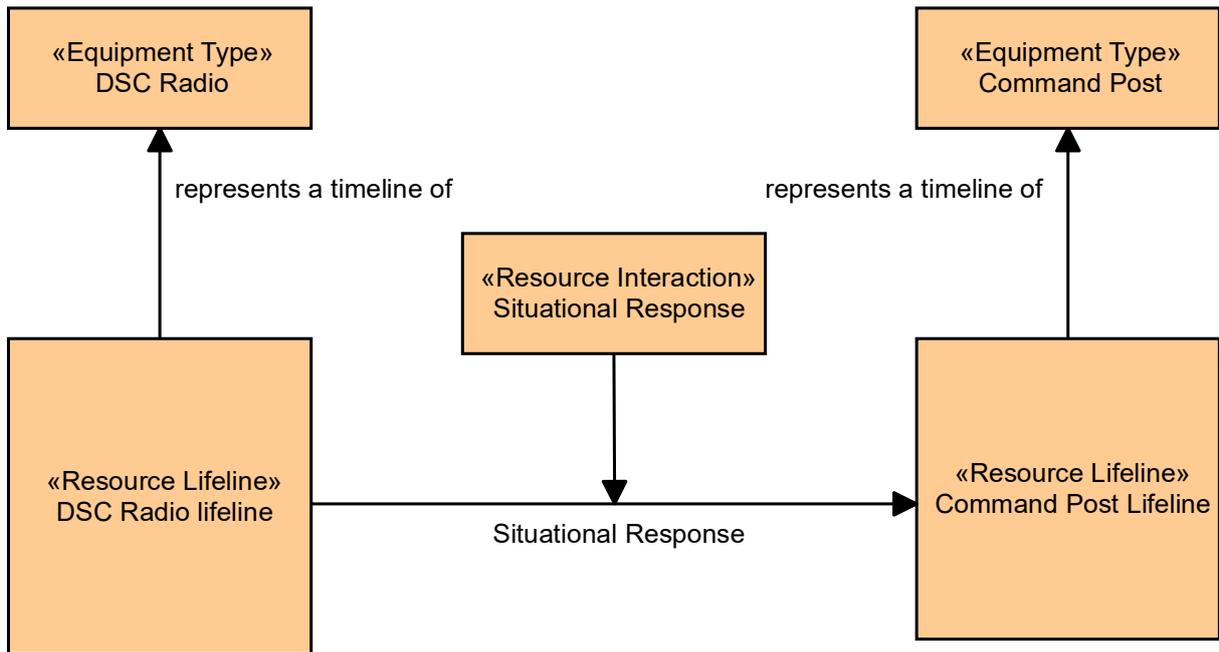
Scope
<ul style="list-style-type: none"> • Shall contain Resource Lifelines representing Physical Active Resources. • Shall contain the Resource Interactions between Resource Lifelines. • May show Resource Events that trigger Resource Interactions. • May show Resource Functions related to Resource Interactions.

Representation
<ul style="list-style-type: none"> • Topological (connected shapes). • Sequence Diagram.

38.1 Cut-out



38.2 NAFv4 Compliant Example

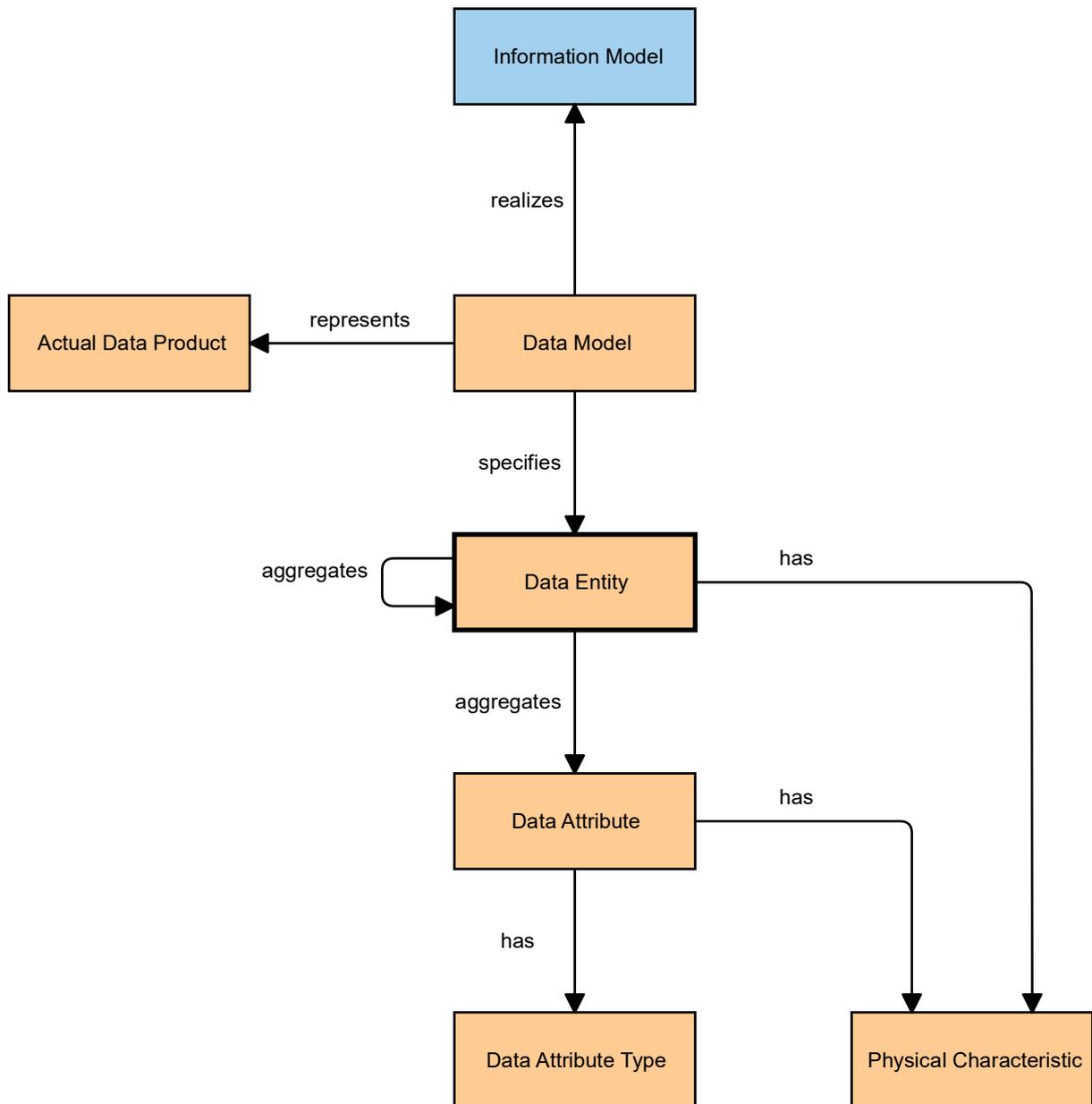


39 P7 - DATA MODEL

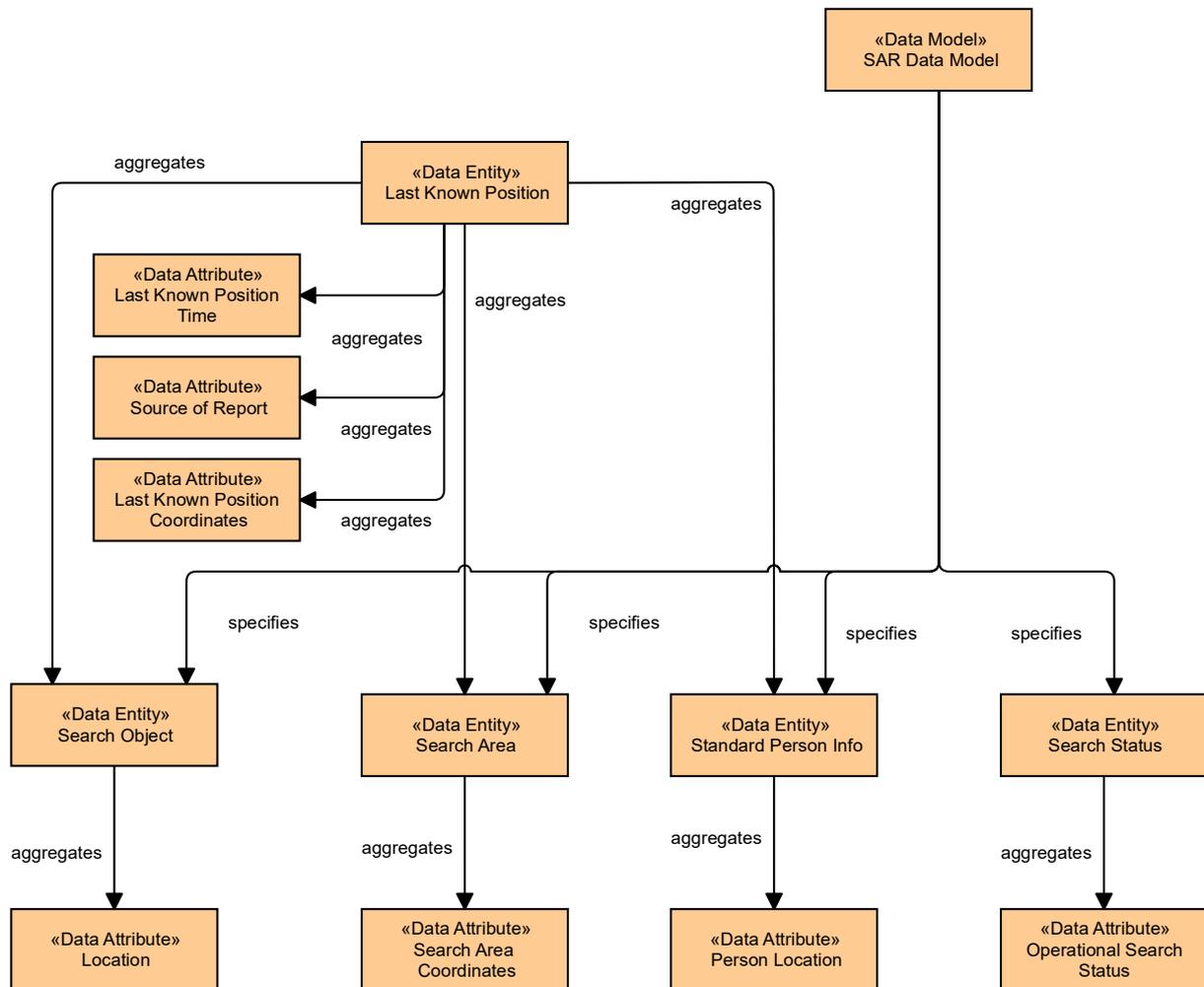
Purpose	Usage & Concerns Addressed
The P7 Viewpoint is concerned with the identifying Data Products, Data Entities and their relationships.	<ul style="list-style-type: none"> • System Design. • Data Schema Design. • Message / Protocol Specification. • Data Architecture. • Database Design. • Specifying the data elements exchanged between systems • Security Modelling.

Scope
<ul style="list-style-type: none"> • Shall contain Data Entities relevant for the entity of interest. • May show the Data Model that specifies the Data Entities. • May show Data Attributes that are part of the Data Entity. • May show groupings of Data Entities. • May show Data Attribute Types relating to Data Attributes. • May show Physical Characteristics of Data Attributes and Data Entities. • May trace Data Models to Information Models.

Representation
<ul style="list-style-type: none"> • Formal text data modelling language. • Topological (connected shapes). • Class diagram. • Entity-Relation diagram

39.1 Cut-out

39.2 NAFv4 Compliant Example



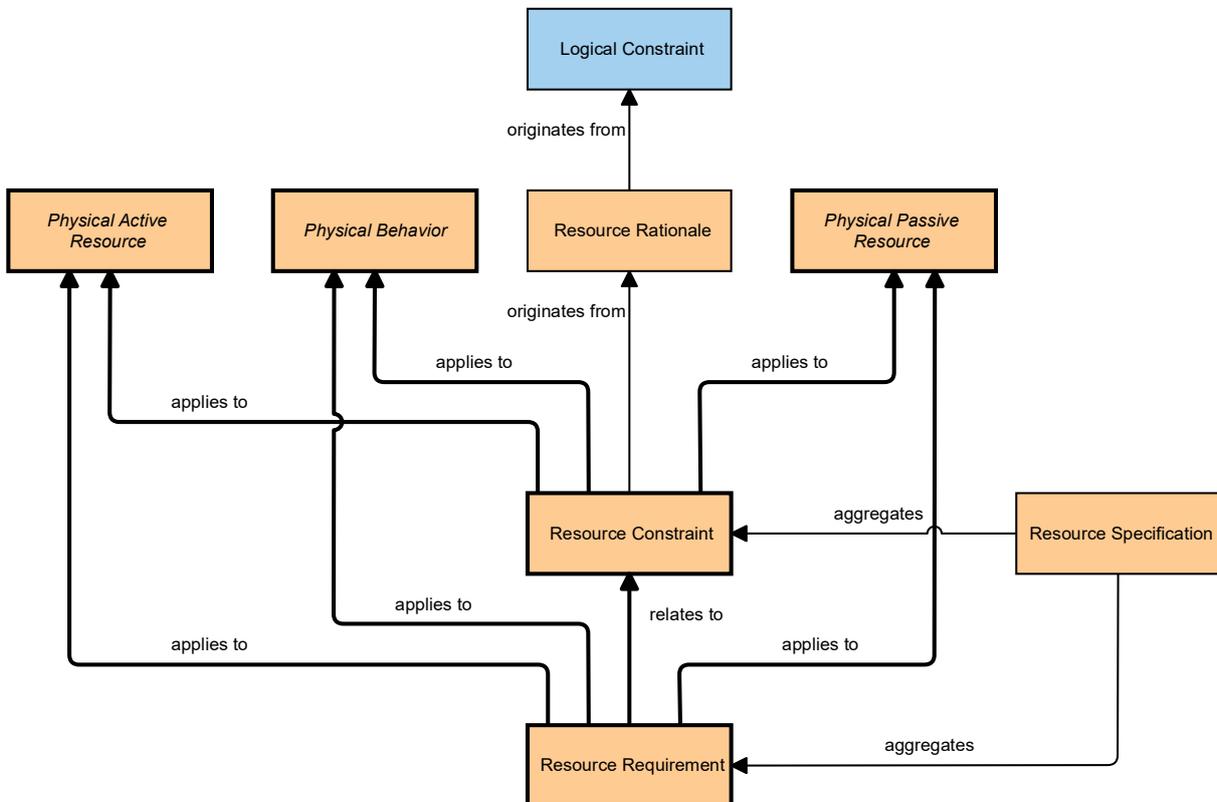
40 P8 - RESOURCE CONSTRAINTS

Purpose	Usage & Concerns Addressed
The P8 Viewpoint is concerned with Identification and description of Resources Constraints and Resource Requirements	<ul style="list-style-type: none"> • Non-Functional Requirements. • Safety Cases. • Definition of implementation logic. • Identification of resource constraints.

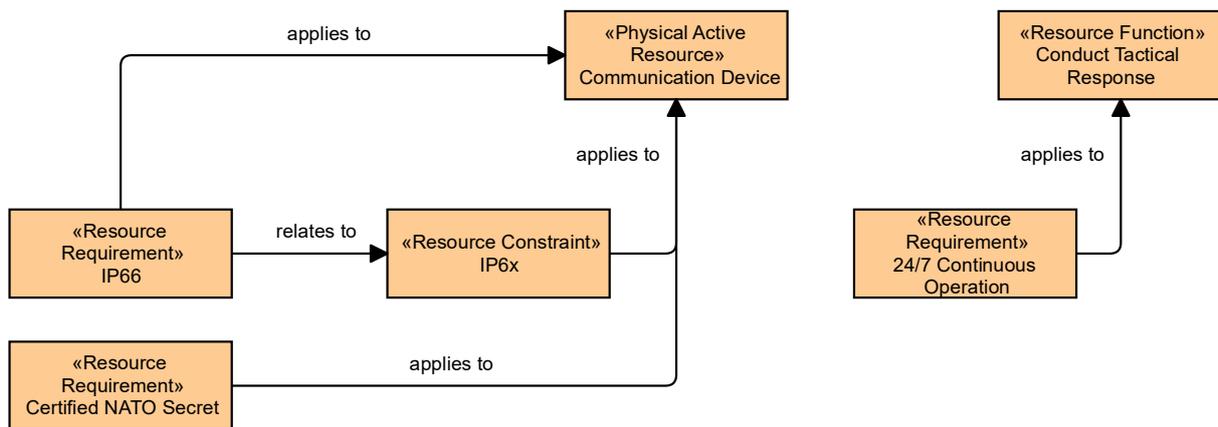
Scope
<ul style="list-style-type: none"> • Shall contain Resource Requirements that relate to Resource Constraints relevant for the entity of interest. • Shall contain Physical Active Resources, Physical Behaviour and Physical Passive Resources that these Resource Constraints or Resource Requirements apply to. • May show Resource Specifications that group Resource Constraints and Resource Requirements. • May show Resource Rationale relating to Resource Constraints. • May trace Resource Rationale to Logical Constraints.

Representation
<ul style="list-style-type: none"> • Text Document • Tabular. • Parametric diagram.

40.1 Cut-out



40.2 NAFv4 Compliant Example



40.3 Notes on the Example

Requirements Engineering is a continuous activity, any P8 output would therefore be a snapshot and be subject to change.

IP6x denotes the constraint, the general Ingress Protection standard family, and implies for a this communication device it needs to be dust-tight (the highest rating). *IP66* defines a specific, compliant **Resource Requirement** derived from that constraint.

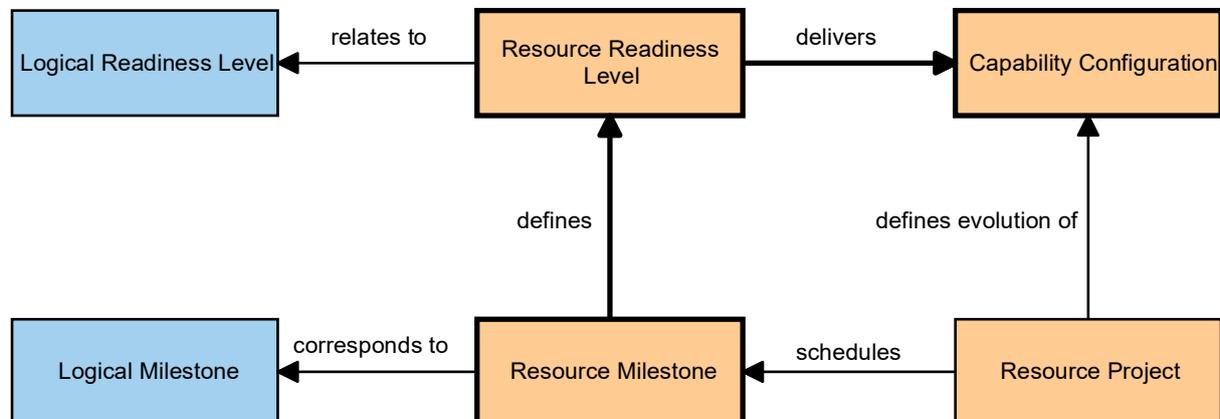
41 PR - RESOURCE ROADMAP

Purpose	Usage & Concerns Addressed
The Pr Viewpoint is concerned with Resource Roadmaps with Resource Milestones related to the evolution of Capability Configurations	<ul style="list-style-type: none"> • Product Lifecycle Management. • Version Control. • Release Scheduling. • Development of incremental acquisition strategy. • Configuration Management. • Planning technology insertion.

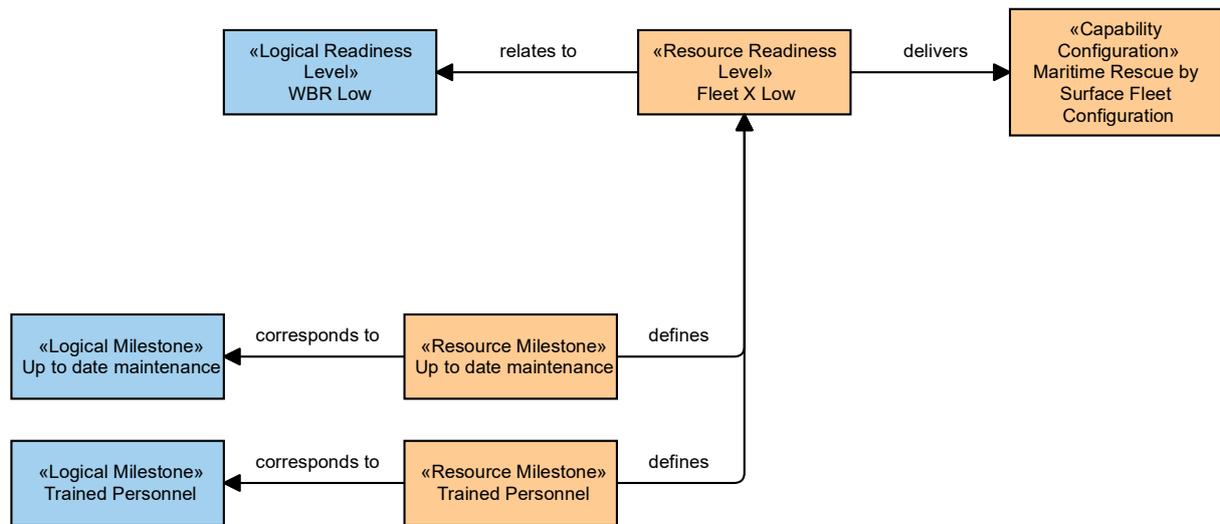
Scope
<ul style="list-style-type: none"> • Shall contain Resource Readiness Levels that Resource Milestones define. • Shall contain Capability Configurations that the Resource Readiness Level delivers. • May show Resource Projects and how they schedule Resource Milestones and define the evolution of the Capability Configuration. • May trace Resource Readiness Levels to Logical Readiness Levels. • May trace Resource Milestones to Logical Milestones.

Representation
<ul style="list-style-type: none"> • Timeline view. • Herringbone style diagram. • Augmented chart in style of a Gantt Chart.

41.1 Cut-out



41.2 NAFv4 Compliant Example



41.3 Notes on the Example

This example, similar to other roadmap viewpoints in NAFv4, is how to model the scenario and is important for the understanding of the evolution of the architecture over time and its impact on decisioning. However, the specific representation of the roadmap for the benefit of non-architectural stakeholder readership should be as described in the content of the 'Representation' section of this viewpoint definition.

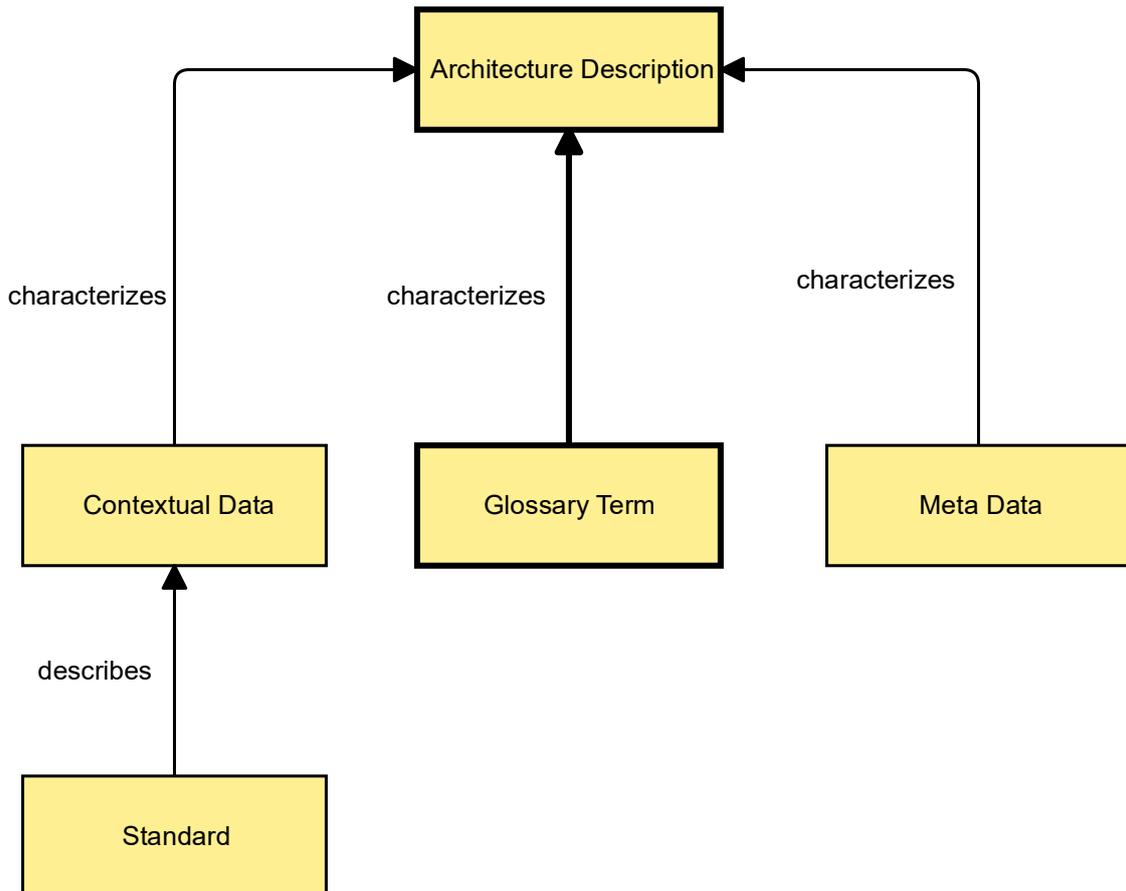
42 A1 - METADATA DEFINITIONS

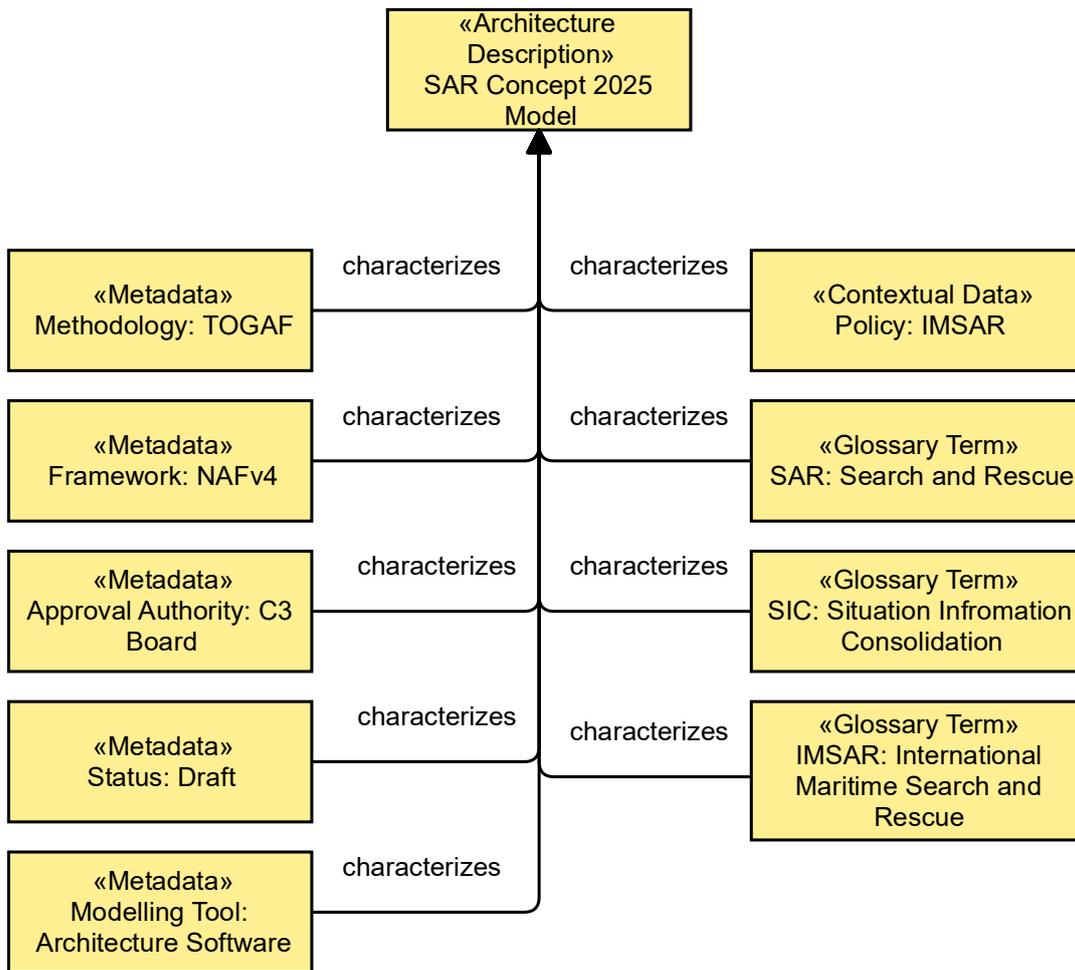
Purpose	Usage & Concerns Addressed
The A1 is concerned with terminology and metadata of different kinds regarding the architecture.	<ul style="list-style-type: none"> • Architecture element discovery. • Security and data protection Management. • Summary and Overview. • Setting up a glossary for the architecture. • Managing architecture metadata.

Scope
<ul style="list-style-type: none"> • Shall contain the Glossary Terms relevant for the Architecture Description. • May show metadata that characterizes the Architecture Description. • May show Contextual Data that characterizes the Architecture Description. • May show Standards that describe the Contextual Data.

Representation
<ul style="list-style-type: none"> • Tabular. • Text.

42.1 Cut-out



42.2 NAFv4 Compliant Example

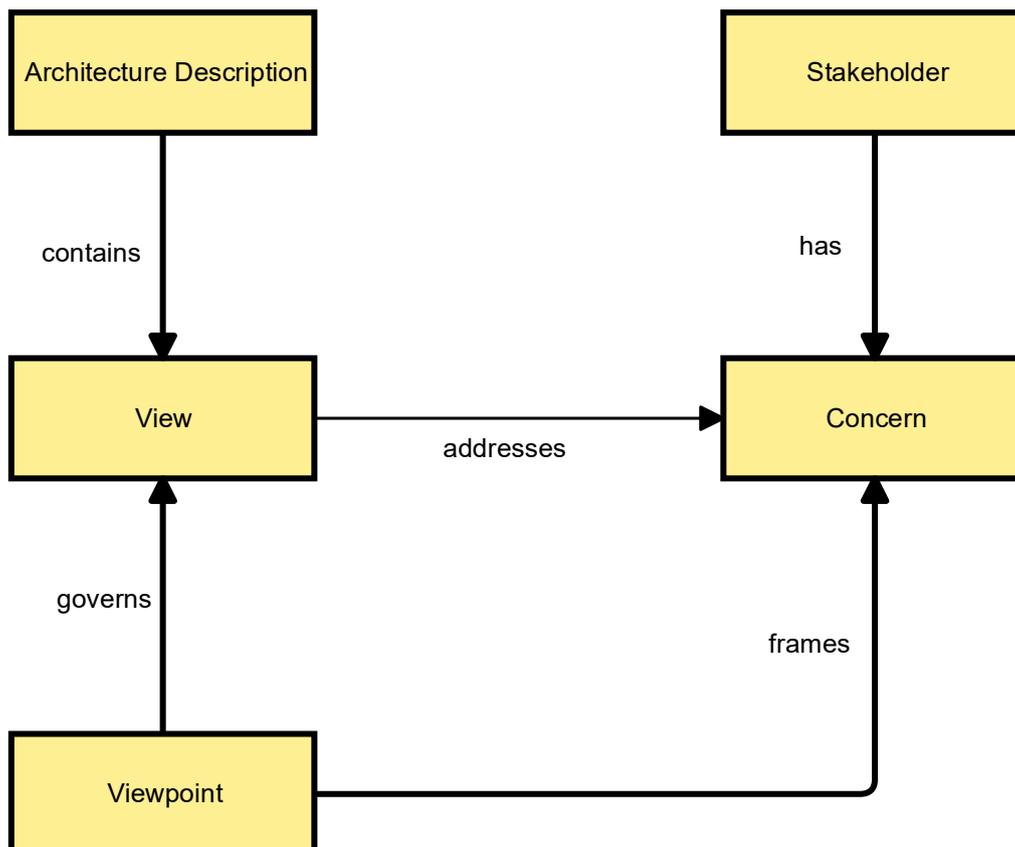
43 A2 - ARCHITECTURE PRODUCTS

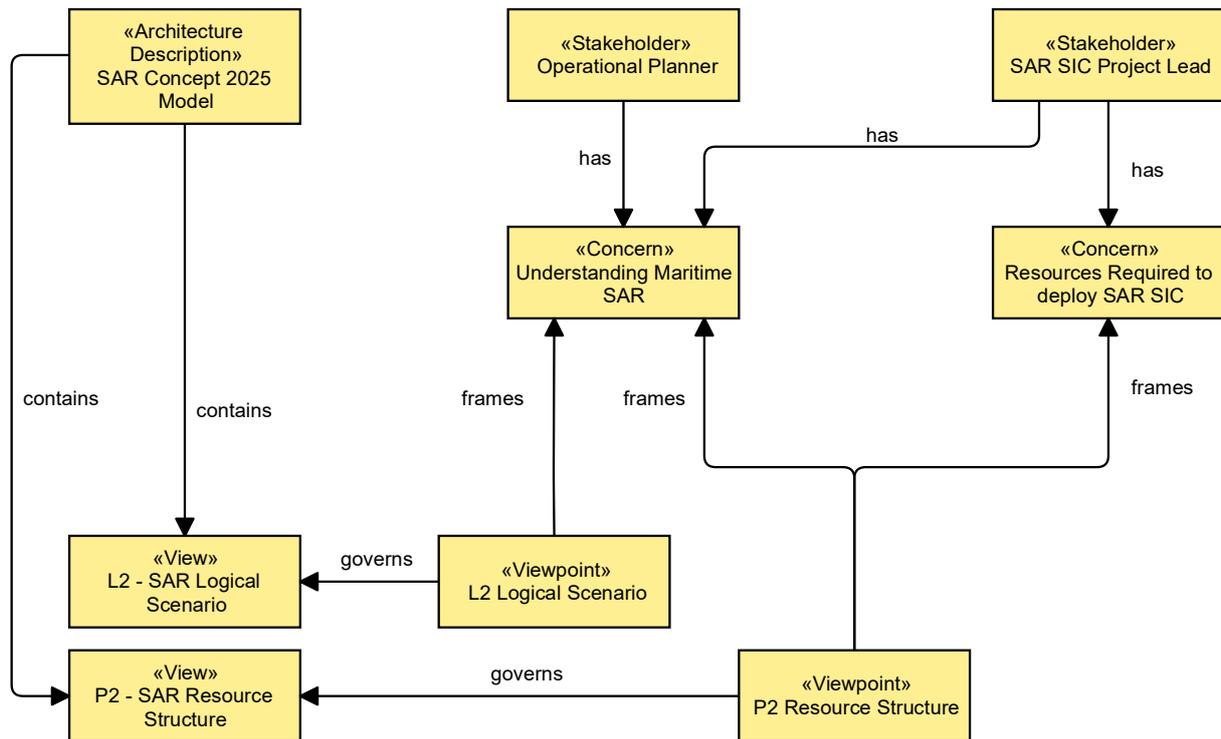
Purpose	Usage & Concerns Addressed
The A2 Architecture Products Viewpoint is concerned with the Architecture Work Products that describe or accompany an Architecture.	<ul style="list-style-type: none"> • Architecture Content. • Stakeholder Management. • Summarizing an Architecture. • Navigating an Architecture.

Scope
<ul style="list-style-type: none"> • Shall contain the Views that make up the Architecture Description and which Viewpoints those Views conform to. • Shall contain Stakeholders, their concerns and the Viewpoints that frame the Concerns. • May show Views that directly address a concern.

Representation
<ul style="list-style-type: none"> • NAF grid representation. • Other representations suitable for the architect.

43.1 Cut-out



43.2 NAFv4 Compliant Example

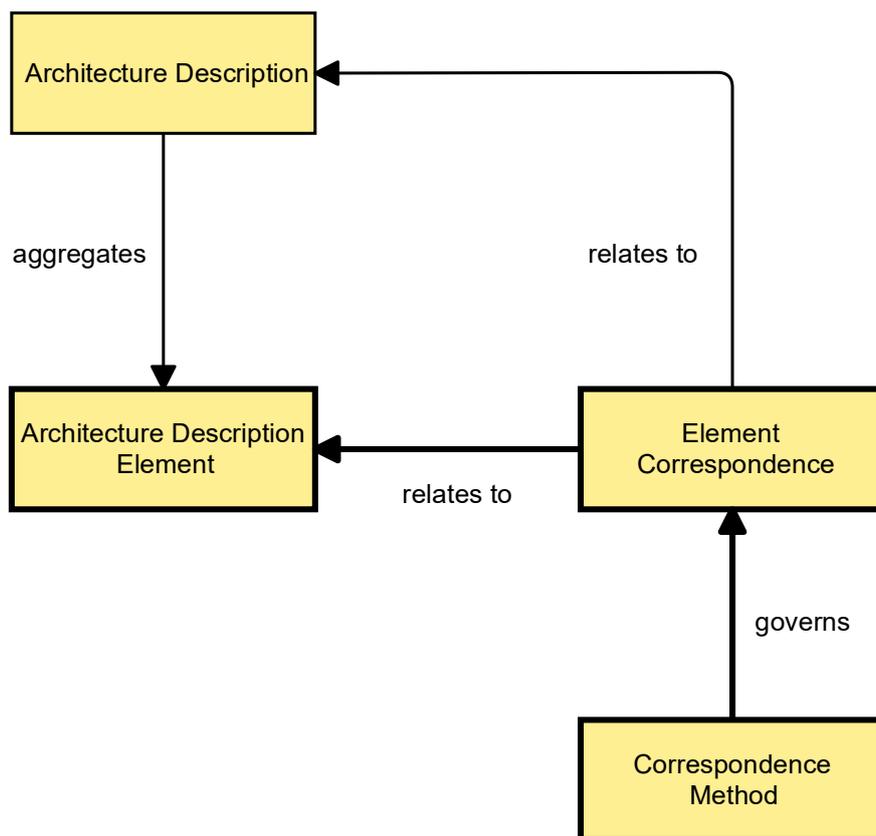
44 A3 - ARCHITECTURE CORRESPONDENCE

Purpose	Usage & Concerns Addressed
The A3 Viewpoint is concerned with implementing correspondence between elements of Architecture Descriptions (AD).	<ul style="list-style-type: none"> • Traceability / dependencies between architectures. • Re-Use of Architectures. • Dependency analysis across architectures.

Scope
<ul style="list-style-type: none"> • Shall contain all relevant Architecture Description Elements. • Shall contain Element Correspondences that relate to Architecture Description Elements and Correspondence Methods that govern them. • May show how Architecture Descriptions relate to Architecture Description Elements. • May show how Architecture Descriptions relate to Element Correspondences.

Representation
<ul style="list-style-type: none"> • Tabular. • Diagram.

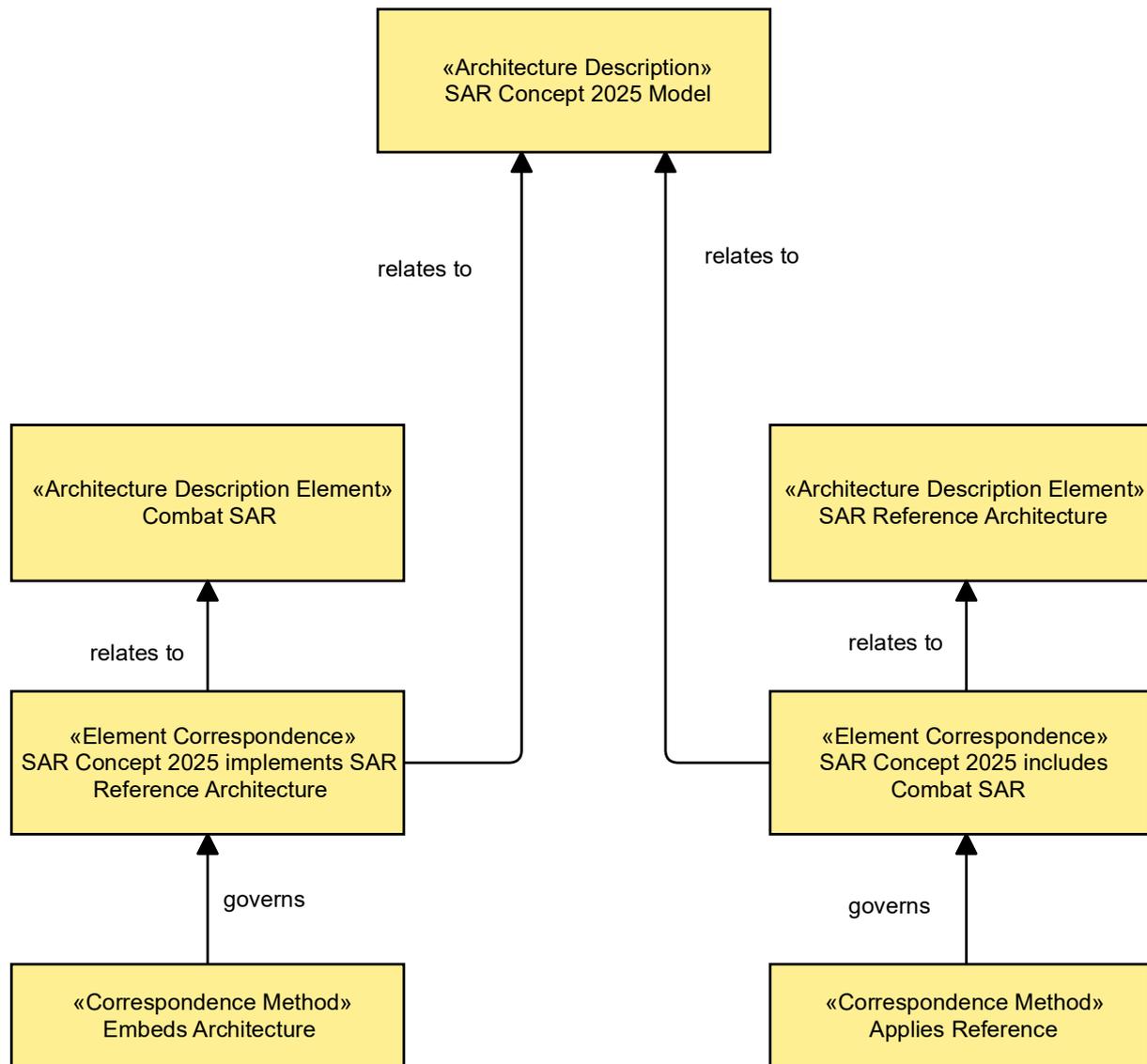
44.1 Cut-out



44.2 Notes on the Cut-out

The concepts of architecture correspondence presented here are derived from ISO/IEC/IEEE 42010.

An Architecture Description element can have any level of granularity, ranging from a component of a view to a complete architecture description.

44.3 NAFv4 Compliant Example

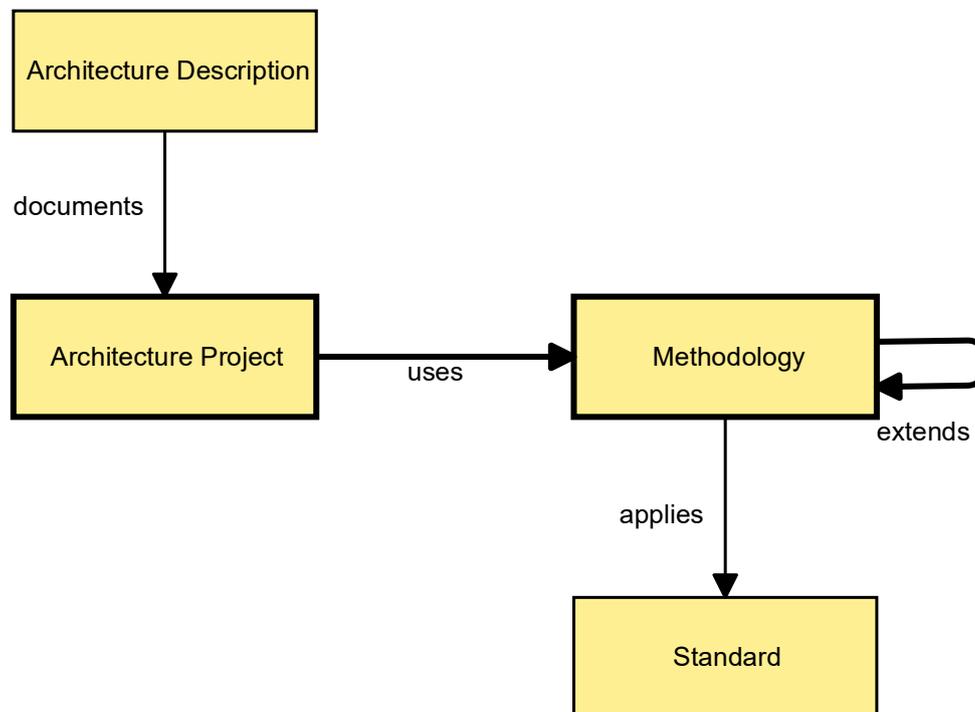
45 A4 - ARCHITECTURE METHODOLOGY

Purpose	Usage & Concerns Addressed
The A4 is concerned with methodology used in developing the architecture.	<ul style="list-style-type: none"> • Architecture Management & Review. • Architecture project management.

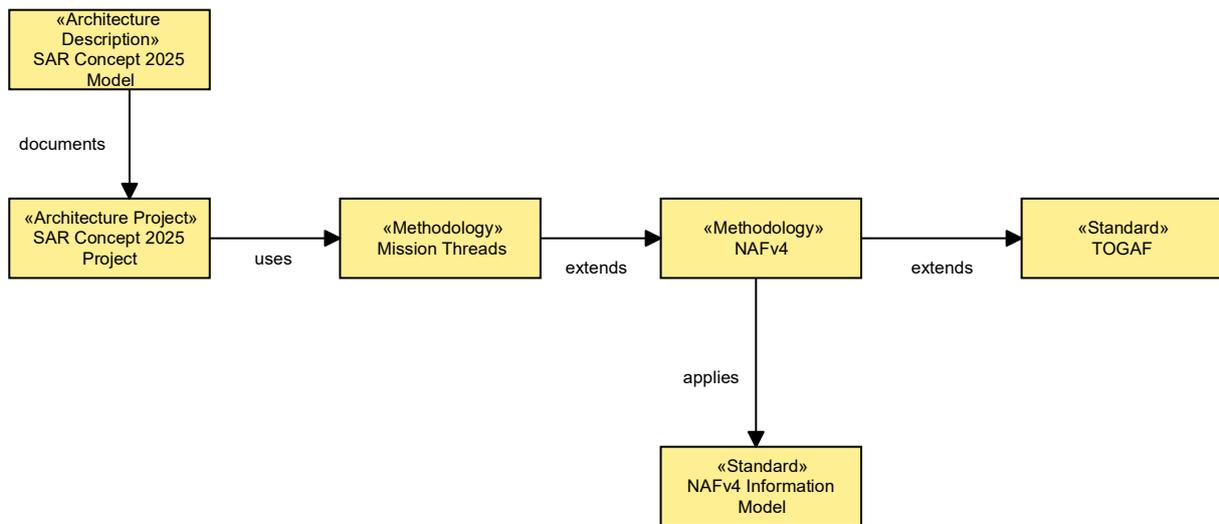
Scope
<ul style="list-style-type: none"> • Shall contain the Architecture Projects and the Methodologies it uses and Methodology extensions. • May show the Standards that apply to the Methodology. • May show the Architecture Description that document the projects.

Representation
<ul style="list-style-type: none"> • Tabular. • Text Document.

45.1 Cut-out



45.2 NAFv4 Compliant Example



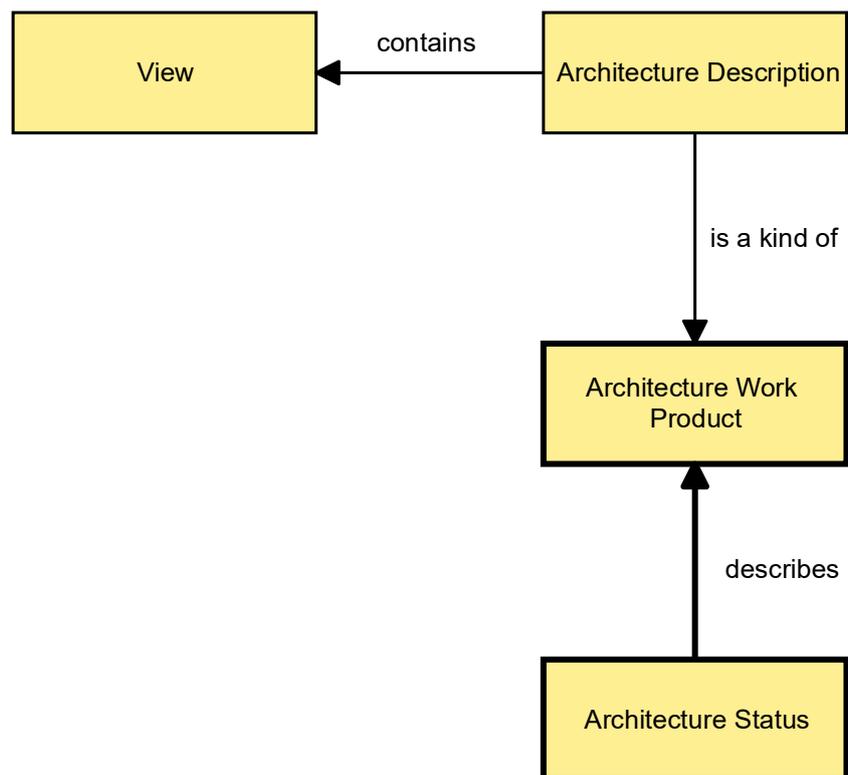
46 A5 - ARCHITECTURE STATUS

Purpose	Usage & Concerns Addressed
The A5 Viewpoint is concerned with usability and approval status of the Architecture Work Products which contribute to development and usability of the Architecture.	<ul style="list-style-type: none"> • Architecture Management & Review. • Architecture project management. • Release scheduling.

Scope
<ul style="list-style-type: none"> • Shall contain Architecture Work Products with their Architecture Status. • May show Architecture Descriptions that relate to Architecture Work Products. • May show Views that are contained in Architecture Work Products or Descriptions.

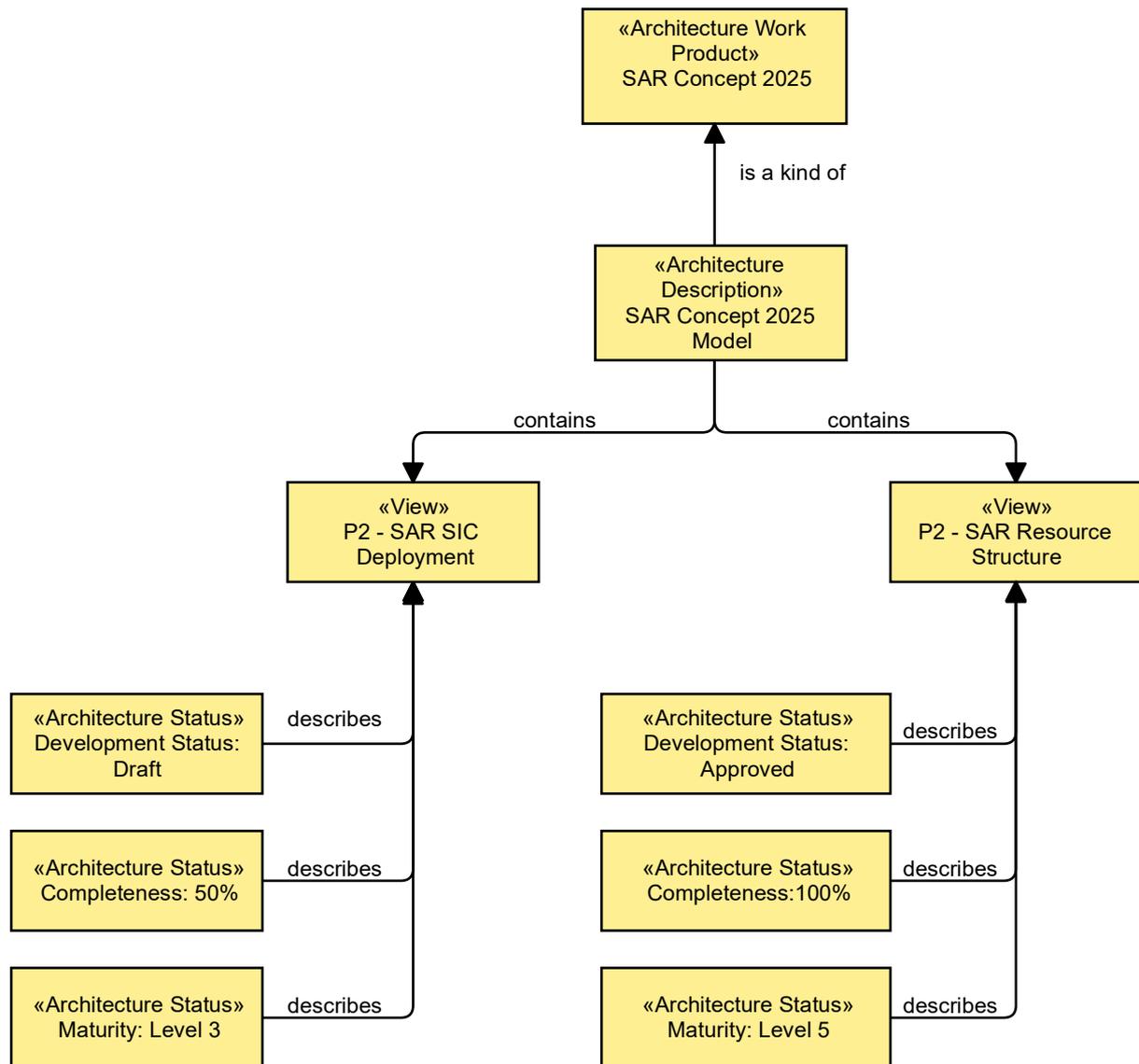
Representation
<ul style="list-style-type: none"> • Tabular. • Text Document.

46.1 Cut-out



46.2 Notes on the Cut-out

The history and roadmap of an Architecture status can be covered in the Ar Viewpoint.

46.3 NAFv4 Compliant Example

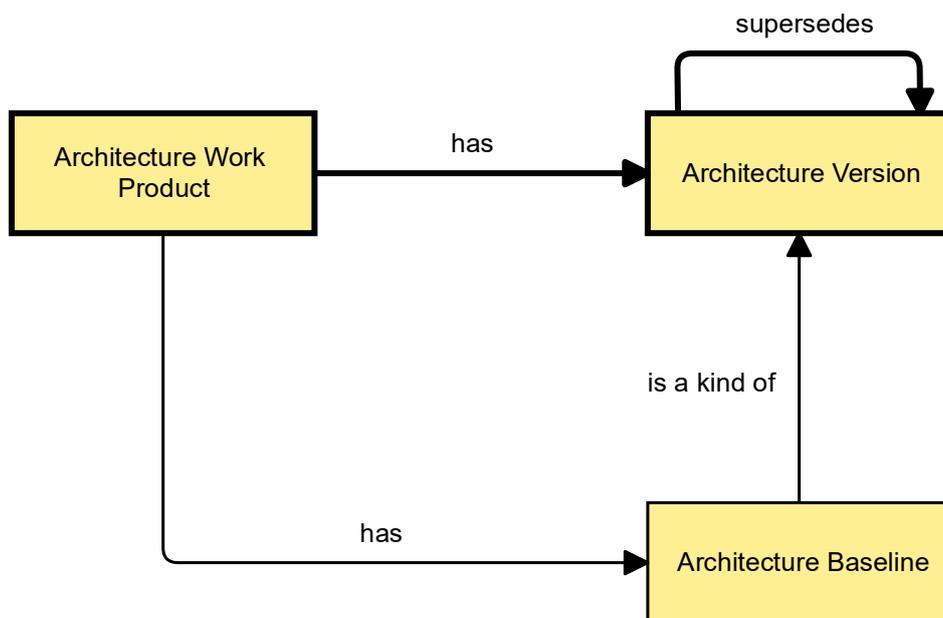
47 A6 - ARCHITECTURE VERSIONS

Purpose	Usage & Concerns Addressed
The A6 Architecture Versions Viewpoint is concerned with version history of the Architecture Work Products.	<ul style="list-style-type: none"> • Architecture Management. • Configuration Control of Architectures.

Scope
<ul style="list-style-type: none"> • Shall contain Architecture Versions and relating Architecture Work Products. • Shall contain the temporal succession of the Architecture Work Product Versions. • May show Architecture Baselines relating to Architecture Work Products and Architecture Versions.

Representation
<ul style="list-style-type: none"> • Tabular. • Text document.

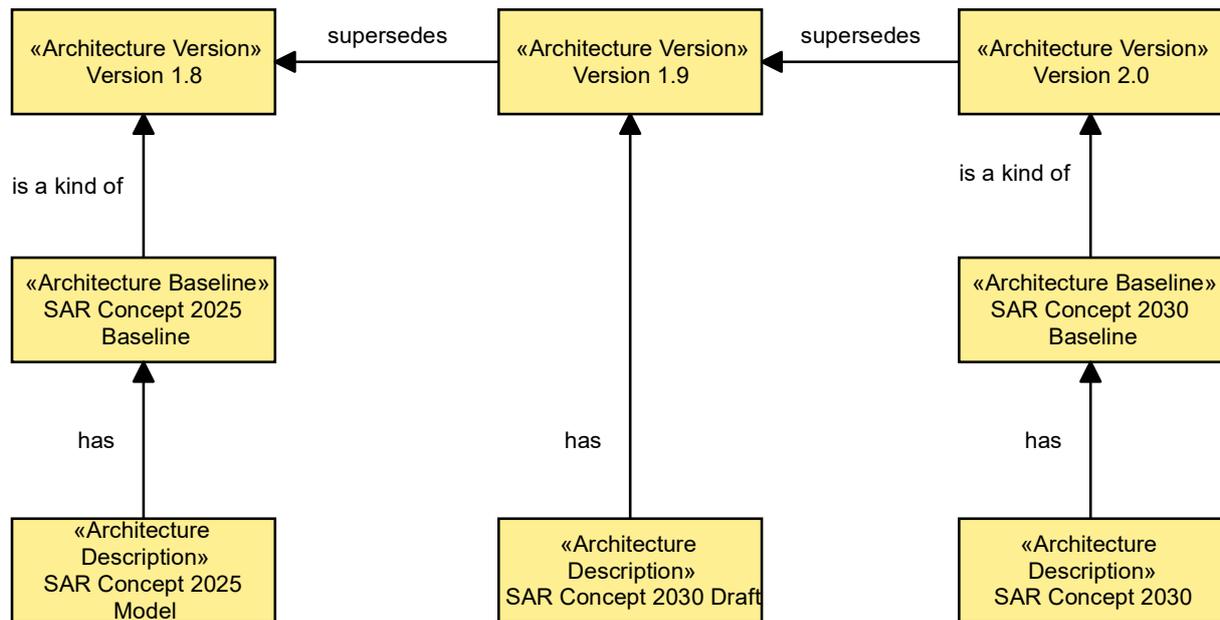
47.1 Cut-out



47.2 Notes on the Cut-out

In the context of NAFv4, a baseline is considered a coherent set of versions of Architecture Work Products.

47.3 NAFv4 Compliant Example



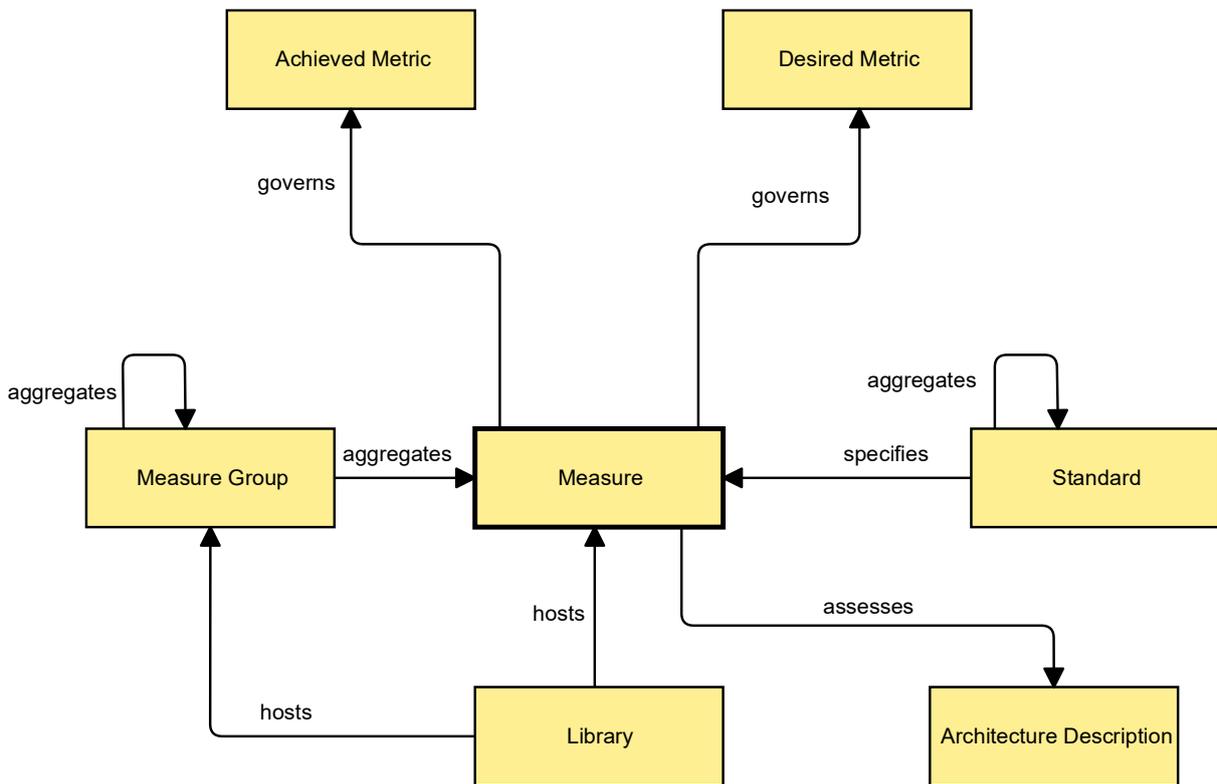
48 A7 - ARCHITECTURE METADATA

Purpose	Usage & Concerns Addressed
The A7 is concerned with the definition of Measures and associated criteria supporting architecture assessment.	<ul style="list-style-type: none"> • View discovery. • Architecture discovery. • Quality assurance. • Architecture element attributes.

Scope
<ul style="list-style-type: none"> • Shall contain Measures relevant for the architecture assessment. • May show Measures organized into Measure Groups, and their groupings. • May show Libraries hosting Measure Groups and Measures. • May show Measures to Standards and the Standards' groupings. • May show Desired Metrics and Achieved Metrics that the Measures govern.

Representation
<ul style="list-style-type: none"> • Tabular. • Diagram.

48.1 Cut-out

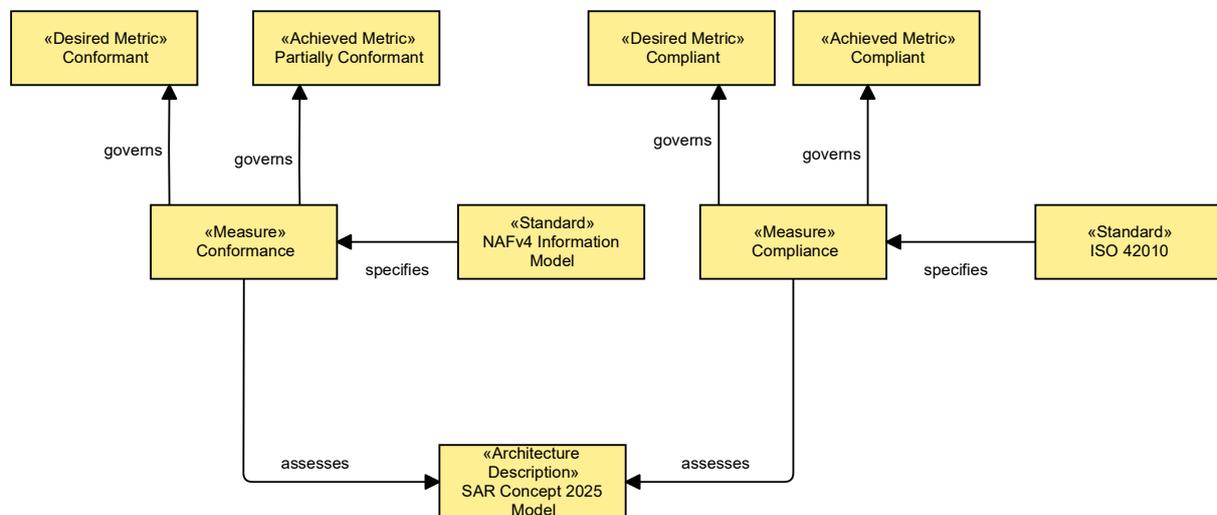


48.2 Notes on the Cut-out

A Measure can be defined with:

- name.
- description.
- data type e.g. float.
- measure unit e.g. km or rounds per minute.
- reference to standard specifying the measurement process.
- reference to standard specifying the mathematics behind the measure unit.

48.3 NAFv4 Compliant Example



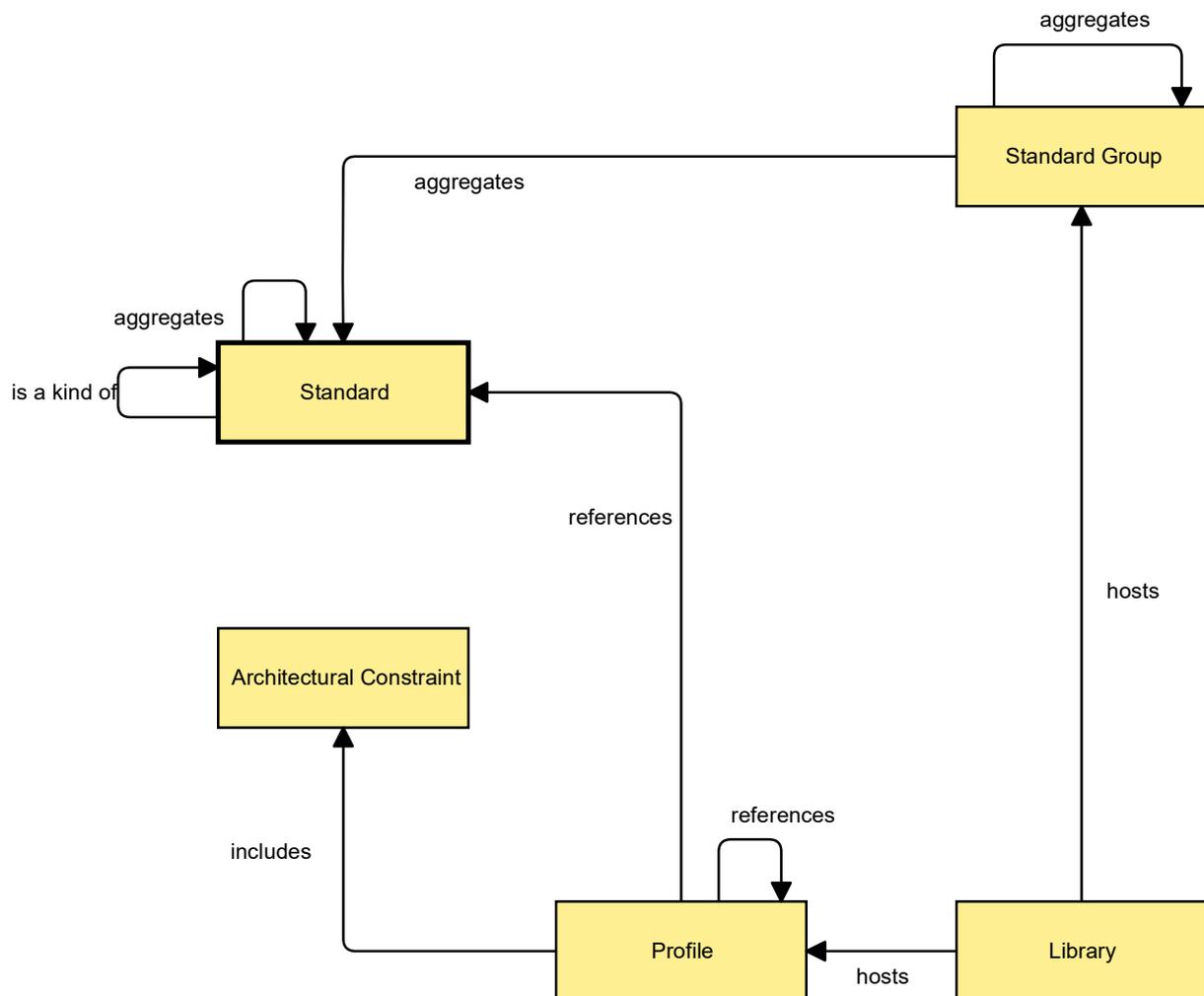
49 A8 - ARCHITECTURE STANDARDS

Purpose	Usage & Concerns Addressed
<p>The A8 Standard Viewpoint specifies the Profiles, Standards and Constraints (rules, policy and guidance) that are applicable to aspects of the architecture. These Standards may be traced to elements elsewhere in the architecture to indicate that those elements conform to the applicable NATO and/or international Standards.</p>	<ul style="list-style-type: none"> • 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.

Scope
<ul style="list-style-type: none"> • Shall contain the Standards used throughout the architecture. • May show Standards in Standard Groups, and their groupings. • May show Profiles which reference Standards and other Profiles. • May show Architectural Constraints that the Profiles include. • May show Libraries hosting Profiles and Standard Groups.

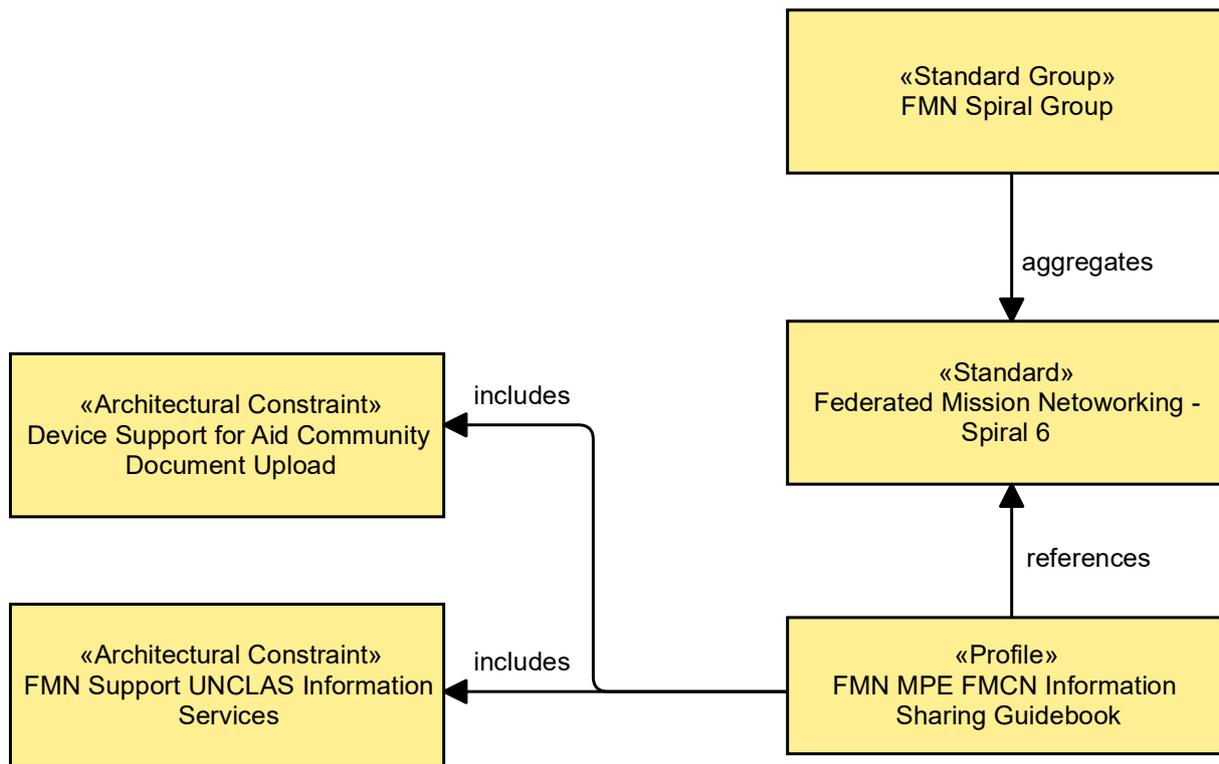
Representation
<ul style="list-style-type: none"> • Tabular. • Diagram.

49.1 Cut-out



49.2 Notes on the Cut-out

The standards here need not be purely technical; they may also relate to language, processes, business or military doctrine, best practice, or legislation. Because the period between concept and fielded capability can be lengthy, it is often necessary to reference standards that, while not yet ratified, are expected to influence future capabilities. These could include forthcoming legislative changes, environmental regulations, or safety standards. Referencing emerging standards in this way allows architects to anticipate future dependencies and reduce the risk of design obsolescence.

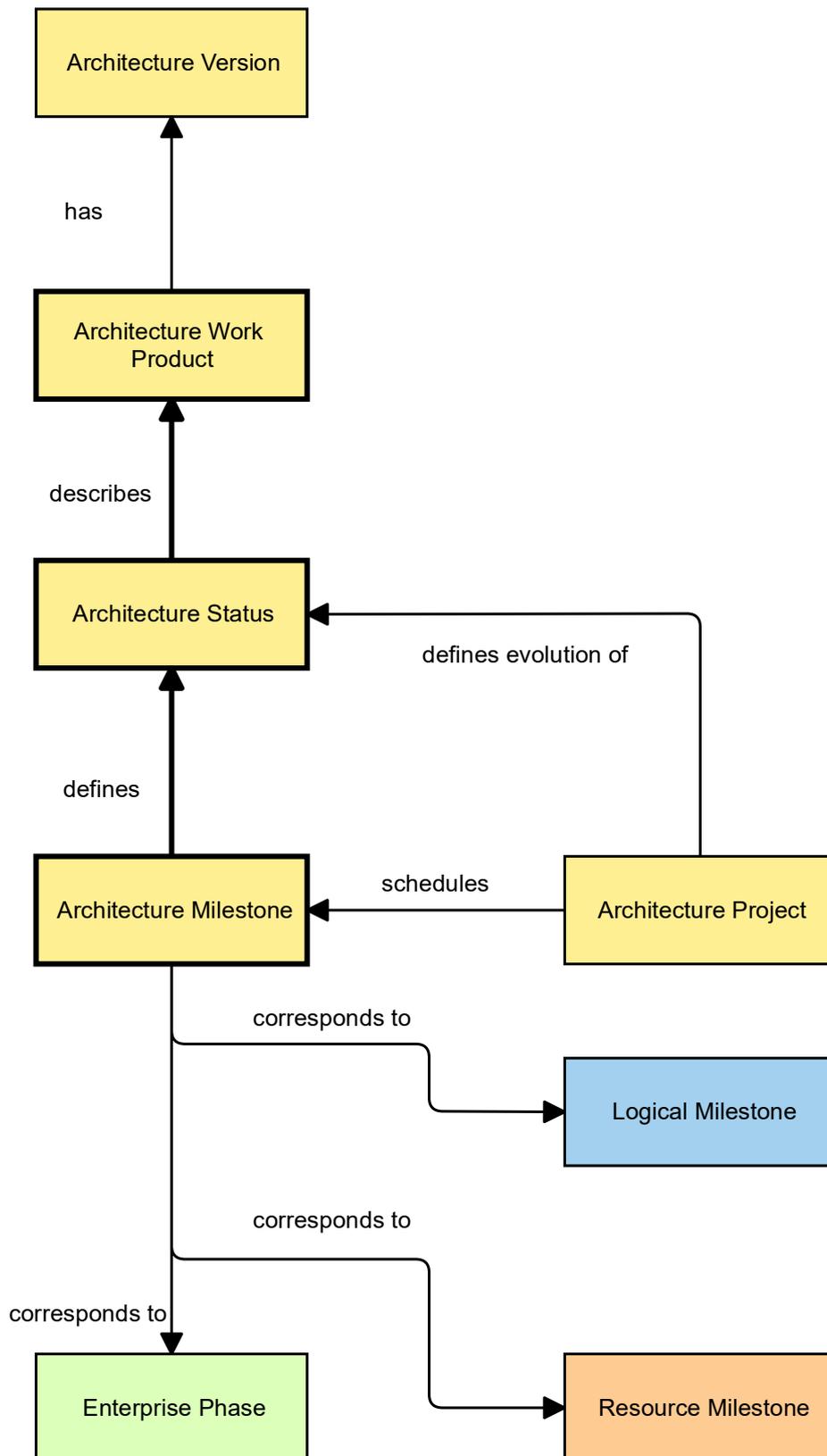
49.3 NAFv4 Compliant Example

50 AR - ARCHITECTURE ROADMAP

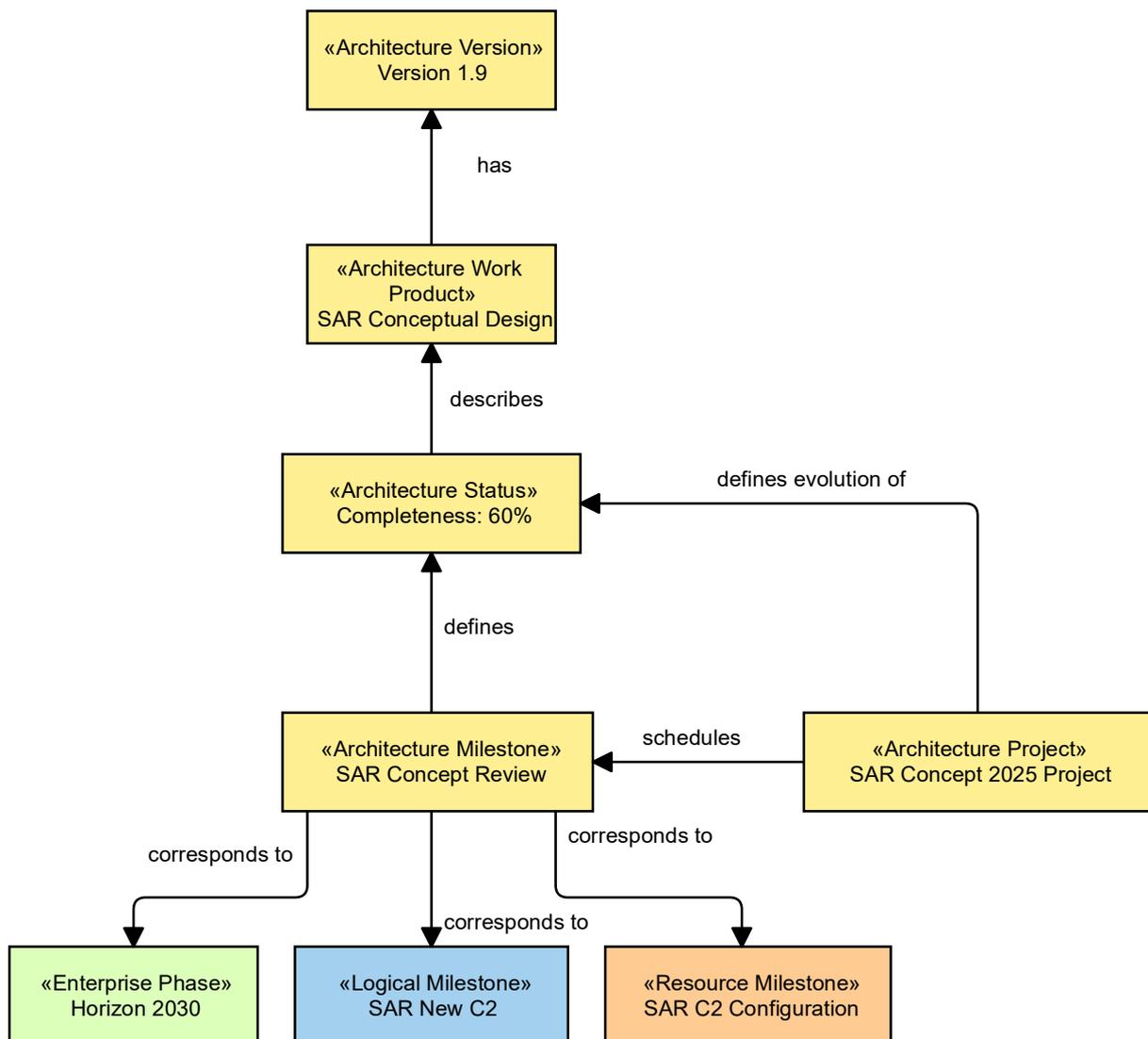
Purpose	Usage & Concerns Addressed
The Ar Architecture Roadmap Viewpoint is concerned with the evolution timeline for an Architecture to meet the goals of the enterprise.	<ul style="list-style-type: none"> • Architecture Project Management. • Developing architectures.

Scope
<ul style="list-style-type: none"> • Shall contain Architecture Statuses of Architecture Work Products. • Shall contain Architecture Milestones that define the Architecture Status. • May show Architecture Projects that schedule a Milestone and define the evolution of an Architecture Status. • May show Architecture Versions of the Architecture Work Product. • May trace Architecture Milestones to other row-specific Milestones they correspond to.

Representation
<ul style="list-style-type: none"> • Timeline view. • Augmented chart in style of a Gantt Chart.

50.1 Cut-out

50.2 NAFv4 Compliant Example



50.3 Notes on Example

This example, similar to other roadmap viewpoints in NAFv4, is how to model the scenario and is important for the understanding of the evolution of the architecture over time and its impact on decisioning. However, the specific representation of the roadmap for the benefit of non-architectural stakeholder readership should be as described in the content of the 'Representation' section of this viewpoint definition.

CHAPTER 4 - META-MODEL

1 METAMODEL

1.1 Introduction

- 1.1.1 Chapter 4 of the NATO Architecture Framework identifies the meta-models to be used for creating NAFv4 compliant architectures.
- 1.1.2 ISO 42010 defines the term meta-model as something that;

“presents the Architecture Description (AD) elements that comprise the vocabulary of a model kind”.

- 1.1.3 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?
- 1.1.4 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)®.
- 1.1.5 The NATO Interoperability Standards and Profiles (NISP) contains the versions of ArchiMate® and UAF DMM® required for compliance.

1.2 ARCHIMATE®

- 1.2.1 ArchiMate® is an open and independent modelling 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.
- 1.2.2 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.

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

- 1.3.1 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.
- 1.3.2 Since scope and expressiveness of the UAF DMM exceed the current needs of NAFv4 and some of the mapped viewpoints differ between NAFv4 and UAF, the use of UAF in

NAFv4 is based on a subset of the UAF DMM described in a separate guideline document.

- 1.3.3 Architectures implemented using the full UAF DMM are fully compliant to NAFv4 when covering the corresponding viewpoints. To ensure further compliance, the additional parts of the UAF DMM must first be used if extending the UAF DMM based NAFv4 metamodel.

CHAPTER 5 – GLOSSARY

1 GLOSSARY

Name	Description
Achieved Metric	A value for a [Metric] observed or measured for a specific Entity of Interest.
Actor	An entity or party that actively performs or contributes to an activity, action or task.
Actual Data Product	A tangible, identifiable instance of a [Data Product Type].
Actual Energy	A concrete and deliverable instance of energy. Examples: Red Diesel, AC 220V 50Hz.
Actual Equipment	An identifiable instance of an [Equipment Type]. Example: The Challenger tank with the plate number H-235.
Actual Infrastructure	An identifiable instance of a [Infrastructure Type].
Actual Material	A concrete and tangible piece of material. Examples: Monkey 47 (Gin), HP Inkjet Cartridge Type 305, Inconel-718+ (nickel alloy).
Actual Organization	An identifiable instance of an [Organization Type]. Example: Charite Hospital in Berlin.
Actual Person	An identifiable instance of an [Actor Type]
Actual Resource Function	A [Function] specified in the context of an Active Resource (Physical).
Actual Technology	An identifiable instance of a [Technology Type].
Architectural Constraint	A limitation, restriction or boundary condition that a solution to a problem must satisfy.
Architecture Baseline	The approved version of an [Architecture Work Product] that can be changed only through formal change control procedures and is used as a basis for comparison.
Architecture Description	A work product used to express an [Architecture] {ISO 42010:2022}.
Architecture Description Element	An identified or named part of an [Architecture Description] {ISO/IEEE/IEC 42010}.
Architecture Milestone	A [Milestone] in a project which predicts or delivers or decommissions one or more [Architecture]s.
Architecture Project	An activity for the delivery of an [Architecture].
Architecture Status	The current state or shape that an [Architecture] is in.
Architecture Version	An identifiable form of an [Architecture] differing in certain respects from an earlier form.
Architecture Work Product	A tangible outcome of architectural work.
Assessment Criteria	A specification of what is expected to be fulfilled by the future realization of a [Capability] (example: system or resource).
Behaviour	Peculiar reaction of an entity under given circumstances arising from an internal or external stimulus.
Behaviour Element	An element that describes a behavioural aspect of an [Architecture].
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.
Capability	A solution building block that combines physical resources (people,

Configuration	technology) and implements processes to realize a [Capability].
Capability Dimension	An aspect of a [Capability] relating to a functional domain or cross-sectional area. Examples for Capability Dimensions: Doctrine, Organization, Training, Materiel, Leadership, Personnel, Facilities, Interoperability, Logistics, Information, Concepts.
Capability Group	Two or more [Capability]s forming a complete unit in a composition or collection. Note: A Capability Group can be structured as a taxonomy.
Capability Increment	An Implementation of a [Capability] that delivers specific value at a specific point in time.
Capability Milestone	A [Milestone] in a project or program which predicts or delivers or decommissions one or more [Capability]s.
Capability Project	An activity for the delivery of a capability
Competencies	Skills, knowledge, experience and attitude that make up a [Person] or are required to fulfil a [Post] or [Role].
Concern	A matter of relevance or importance to a stakeholder {ISO 42010:2022}.
Contextual Data	[Data] describing the environment of surrounding things, conditions, or influences upon an entity of interest.
Correspondence Method	Method used to govern the [AD Element Correspondence].
Data	A reinterpretable representation of information in a formalized manner suitable for communication, interpretation, or processing {NATO Definition}.
Data Attribute	A representation of a property of a [Data Entity].
Data Attribute Type	A set of constraints that specify the allowable values of the Attribute on an implementation-specific level. Examples: 64bit-Integer, Unicode String.
Data Entity	An identified unit or piece of [Data].
Data Model	A [Model] describing the [Data] referring to a specific Application Domain. Note: An Data Model typically describes the structure of the [Data] in terms of entities, relationships and their attributes.
Data Product	An item that is elaborated from one or more sources of [Data] to meet a specific purpose. Note: quality assurance aspects are frequently associated to a Data Product which is delivered to a decision maker (e.g. integrity and credibility).
Data Product Type	A particular kind of [Data Product]. Examples: an APP6-A Symbology, a database schema, a CSV format, an XML record.
Desired Metric	A value for a [Metric] required for a specific Entity of Interest.
Effect	The consequence or outcome of an action.
Effectiveness Criteria	An operational criterion of effectiveness or success which is specified for a [Capability] or an [Effect].
Element Correspondence	An identified or named relation between two or more architecture description elements {ISO 42010:2022}.
Enduring Task	A type of template behaviour recognized by an enterprise as being essential to achieving its goals.
Energy	A resource that provides power, strength, force for any kind of purpose.
Energy Concept	A primary form of [Energy] specified at the conceptual level. Examples: kinetic energy, potential energy.
Energy Type	A particular, consumable kind of [Energy]. Examples: Alternate Current, Petrol-based Fuel, Microwave.
Enterprise Phase	A current or future state of the enterprise.
Enterprise Vision	The overall aim of an enterprise over a given period of time.
Equipment	All non-expendable items needed to outfit/equip an individual or organization {NATO Definition}.
Equipment Type	A particular kind of [Equipment]. Examples: Challenger tank, Apache

	helicopter, Javelin anti-tank missile, etc.
Event	Something that happens or might happen which may be significant or noteworthy.
Function	A task, action, or activity that must be accomplished to achieve a desired outcome {ISO/IEC/IEEE 26514:2022}.
Glossary Term	The name and definition of a defined concept that appears within an [Architecture].
Goal	The aim or outcome which a person, group, or organization works towards or strives to achieve.
Information	Knowledge that is exchangeable amongst users, about things, facts, concepts, and so on, in a universe of discourse {ISO/IEC 10746-2:2009}.
Information Attribute	A representation of a property of an [Information Entity].
Information Attribute Type	A set of constraints that specify the allowable values of the Attribute on an implementation-independent level. Examples: Integer, Real, String.
Information Entity	An identified unit or piece of [Information].
Information Model	A [Model] describing the [Information] referring to a specific Domain. Note1: An Information Model typically describes the structure of the [Information] in terms of entities, relationships and optionally their attributes. Note2: an Information Model should be accompanied by a glossary or an ontology describing the semantics of the [Information].
Information Product	An item that is derived from one or more sources of [Information] to meet a specific purpose.
Information Product Concept	An [Information Product] specified at the conceptual level.
Infrastructure	The basic physical and organizational structures and facilities (e.g. buildings, roads, power supplies) needed for the operation of a society or enterprise.
Infrastructure Type	A particular kind of [Infrastructure].
Interaction	A kind of activity or communication which involves two parties.
Library	A collection of artefacts providing information.
Lifeline	A linear representation of the course of an entity's life.
Logical Active Resource	An [Active Resource] specified at the logical level.
Logical Actor	An [Organization] specified at the conceptual level.
Logical Behaviour	[Behaviour] specified at the logical level.
Logical Constraint	A [Constraint] specified at the logical level.
Logical Energy	A category of [Energy] that relates to a specific source. Examples: thermal energy, radiant energy, chemical energy, nuclear energy, electrical energy, motion energy, sound energy, elastic energy or gravitational energy.
Logical Equipment	An [Equipment] specified at the conceptual level.
Logical Event	An [Event] specified at the logical level. Example: Casualty at the point of injury.
Logical Function	A [Function] specified at the logical level. Example: Coffee making.
Logical Interaction	An [Interaction] between two logical elements.
Logical Lifeline	A [Lifeline] specified at the logical level.
Logical Material	A category of [Material] that relates to a specific application domain. Example: Ammunition, engineering material, construction material.
Logical Milestone	A [Milestone] in a project which predicts or delivers or decommissions one or more [Capability Dimension]s.
Logical Mode	A [Mode] specified at the logical level.
Logical Passive Resource	A [Passive Resource] specified at the logical level.

Logical Project	An activity for the delivery of a logical architecture
Logical Rationale	A [Rationale] that is applied at the logical level e.g. because tanks should be transportable by rail or road transport, dimensional limits may have to apply.
Logical Readiness Level	A [Readiness Level] specified at the logical level. Example: The [Readiness Level] of a [Capability Dimension].
Logical Requirement	A [Requirement] identifying user's and/or business expectations. Example: the needs expressed in a User Requirements Document (URD).
Logical Specification	A [Specification] of requirements and constraints identified by the user and/or business.
Logical State	A [State] specified for a logical element. Example: the state of a patient could be unstable.
Logical State Group	Two or more [Logical State]s forming a complete unit in a composition or collection.
Logical Technology	A [Technology] specified at the conceptual level.
Logical Transition	A change from one [Logical State] to another.
Material	Matter from which a thing is made or can be made.
Material Concept	A [Material] specified at the conceptual level. Example: Liquids, solid, gas.
Material Type	A particular kind of [Material]. Examples: Clay, Wood, Textiles, Food.
Measure	A unit used for stating a numerical or categorical attribute of an entity of interest. Examples: meters or inches.
Measure Group	Two or more [Measure]s forming a complete unit in a composition or collection.
Meta Data	[Data] that describe other [Data].
Metric	A criterion or set of criteria stated in quantifiable terms. Example: length, width, weight.
Milestone	A significant stage or event in the progress or development of something.
Mode	A functional condition of an activity or function at a point in time.
Model	A physical, mathematical or otherwise logical representation of a system, entity, phenomenon, or process {NATO Definition}.
Needline	A relationship specifying the need to exchange information between [Node]s.
Node	A placeholder for an aggregate of logical resources and behaviour (including: organizations, equipment, technology, material, information, activities etc) that together perform specific activities and/or functions.
Operational Activity	An element of logical behaviour, specified independently of how it is carried out.
Operational Flow	An exchange of information, material, people, energy or another [Resource] between two [Operational Activity]s.
Organization	An organized group of people with a particular purpose such as business or government department.
Organization Type	A particular kind of [Organization] with specific characteristics. Examples: Driving Licensing Agency, Hospital.
Passive Resource	A [Resource] that is acted upon by an [Active Resource] or exchanged between two [Active Resource]s.
Person	A human as an individual, being used to define the characteristics needed.
Person Type	A particular kind of an [Actor].
Physical Active Resource	An [Active Resource] specified at the physical level.
Physical Behaviour	[Behaviour] specified at the physical level.
Physical Characteristic	Characteristics, constraints or assumptions that apply to the physical storage or transport of [Data Entity]s and/or [Data Attribute]s, regarding for example storage capacity, availability, reliability, electromagnetic compatibility, indexing, redundancy, etc.

Physical Passive Resource	A [Passive Resource] specified at the physical level.
Post	The type of job title or position that a person can fill. Example: Chief Executive Officer, Lawyer, Machine Operator, Cook.
Profile	A set of one or more standards, with one or more constraints and implementation guidance to support development and operation.
Protocol	A kind of [Standard] specifying how two entities communicate or interact with each other.
Rationale	A reason behind a rule or [Constraint]. Examples are rules, policies, doctrine, standards, laws.
Readiness Level	A measured willingness or state of being prepared for something.
Requirement	A statement which specifies an expectation that a solution shall fulfil.
Resource	A stock or supply of money, materials, equipment, personnel, energy, and other assets (e.g. Documentation, Data) that can be drawn on by a person or organization in order to function effectively.
Resource Constraint	A [Constraint] specified at the resource level.
Resource Event	An [Event] specified at the resource level.
Resource Flow	An exchange of data, material, people, energy or another [Resource] between two [Resource]s.
Resource Function	A [Function], specified at the physical level, but independently of the [Physical] Active Resource that will later perform it.
Resource Interaction	An [Interaction] specified at the resource level.
Resource Lifeline	A [Lifeline] specified at the resource level.
Resource Milestone	A [Milestone] in a project which predicts or delivers or decommissions one or more [Resource]s.
Resource Mode	A [Mode] specified at the resource level.
Resource Project	An activity for the delivery of a physical resource
Resource Rationale	A [Rationale] that is applied at the resource level.
Resource Readiness Level	A [Readiness Level] specified at the resource level.
Resource Requirement	Functional and non-functional [Requirement]s describing a system. Example: the requirements expressed in a System Requirements Document (SRD).
Resource Specification	A [Specification] of requirements and constraints identified at the system level. Example: features and behaviour expressed in a System Requirements Specification (SRS).
Resource State	A [State] specified for a [Resource]. Example: a cell phone could be off.
Resource State Group	Two or more [Resource State]s forming a complete unit in a composition or collection.
Resource Transition	A change from one [Resource State] to another.
Roadmap	A plan or strategy that defines a goal or desired outcome and includes the major steps or milestones needed to reach it.
Role	The characteristics by which an entity participates in a process or interaction.
Service	A means of providing value, functionality or a product to a consumer (human or machine) in a standardized and contracted way by hiding associated risks and constraints.
Service Constraint	A [Constraint] specified at the service level.
Service Consumer	An entity which makes use of a [Service].
Service Event	An [Event] specified at the service level.
Service Function	A [Function] specified at the service level.

Service Function Parameter	A constant or variable passed into or out of a [Service Interface] as part of the execution of a [Service Function].
Service Group	Two or more [Service]s forming a complete unit in a composition or collection.
Service Interaction	An [Interaction] between two [Service]s.
Service Interface	A specification of the functional aspects of a [Service]. Example: the operations and/or functions provided by the [Service] and the parameters they use as input or output.
Service Lifeline	A [Lifeline] specified at the service level.
Service Milestone	A [Milestone] in a project which predicts or delivers or decommissions one or more [Service]s.
Service Mode	A [Mode] specified at the service level.
Service Operation	A [Function] or procedure which enables programmatic communication with a [Service] via a [Service Interface].
Service Operation Parameter	A constant or variable passed into or out of a [Service Interface] as part of the execution of a [Service Operation].
Service Policy	A set of rules and constraints that specify the non-functional aspects of a [Service]. Examples: Availability, Reliability, Safety, Security, Useability.
Service Project	An activity for the delivery of a Service
Service Provider	An entity which delivers a [Service].
Service Readiness Level	A [Readiness Level] specified at the service level.
Service State	A [State] specified for a [Service]. Example: a mail service is in maintenance state.
Service State Group	Two or more [Service State]s forming a complete unit in a composition or collection.
Service State Transition	A change from one [Service State] to another.
Specification	An information part that identifies, in a complete, precise and verifiable manner, the requirements, design, behaviour, or other expected characteristics of an entity {ISO 42010:2022}.
Stakeholder	Individual or organization having an interest in an entity or a course of action. Adapted from ISO 15288
Standard	A Standard is a document developed by consensus of different parties and approved by a recognized body with the objective to maximize compatibility, interoperability, safety, repeatability, or quality.
Standard Group	Aggregates Standard elements that that are part of the same [Profile].
Standard Operational Activity	An [Operational Activity] that is a standard procedure specified by policies (e.g. doctrine).
State	A functional condition of a node, operational performer or resource at a point in time.
Strategic Constraint	A limitation, restriction or boundary condition that a solution to a problem must satisfy.
Strategic Rationale	A [Rationale] that is applied at the strategic level.
Structure	The arrangement and organization of interrelated elements in an entity.
Structure and Behaviour	An element that describes a structural or behavioural aspect of an [Architecture].
Structure Element	An element that describes a structural aspect of an [Architecture].
Technology	Machine or equipment developed from the application of scientific knowledge. In the context of the NAF, the term technology is often referring to the IT domain (Examples: Software, Hardware, Firmware, etc).
Technology Type	A particular kind of [Technology].

View	An Information part comprising a portion of an architecture description {ISO 42010:2022}.
Viewpoint	A set of conventions for the creation, interpretation and use of an architecture view to frame one or more concerns {ISO 42010:2022}.

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/IEC/IEEE 24765, 2017	Systems and software engineering – Vocabulary
ISO/IEC 38500, 2015	Information technology -- Governance of IT for the organization
ISO/IEC/IEEE 42010, 2011	Systems and software engineering – Architecture description https://www.iso.org/standard/50508.html
ISO/IEC/IEEE 42020, 2019	Systems and software engineering – Architecture processes https://www.iso.org/standard/68982.html
ISO/IEC/IEEE 42030, 2019	Systems and software engineering – Architecture evaluation https://www.iso.org/standard/73436.html
TOGAF®, 2011	TOGAF Version 9.1®, The Open Group®, 2009-2011 www.opengroup.org/architecture/togaf
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

3 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, Architecture 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
Evans, 2013	David Evans and Mike Wilkinson, UK MOD's perfect storm and the need for Architectural Styles, NATO STO-MP-SCI-254 Symposium on Architecture Assessment for NEC, Tallin, Estonia, 2013
Evans, 2018	David Evans, Styles of Architecting - A smarter approach to architecting the Defense Enterprise, Niteworks white paper, 28 March 2018
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-FEAPCut.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%202008Thoughts_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-
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	

4 DOCUMENT CHANGE

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

4.1.2 As a NATO Tidepedia user please submit any issues by use of the issue tracker linked in the Architecture Body of Knowledge (ABoK), or follow this link:

https://tide.act.nato.int/bugzilla/issues/enter_bug.cgi?product=ACaT%20Documents

Date	Change (Issue Number)
31 Oct 2019	<p>Changed document version from initial to 2019.10. Added reference to Tidepedia issue tracker (4087). Corrected figures 2-3 (4189), 3-11 (4199), 3-3 (4260). Corrected S3 example description (4199). Replaced descriptions of A8 (4210), L4 example (4201), L8 example (4204). Editorial changes (4187, 4190, 4192, 4193, 4194, 4195, 4202, 4208, 4203, 4205, 4196, 4200).</p>
03 Jul 2020	<p>Changed document version to 2020.07. Added the S2 Service Structure viewpoint (4257). Reworked viewpoint examples and example descriptions (4825, 4879). Replaced modeling language specific viewpoint representations by generic ones (4466). Renamed Architecture Meta-Data (A-Row) to Architecture Foundation (4861), changed A7 from Meta-Data to Architecture Compliance (4209), changed descriptions of A1 (5285), A2 (4877, 4881), A4 (4878), A5 (4880), A7 (4209). Changed L7 from Logical Data Model to Information Model, changed P7 from Physical Data Model to Data Model (4876). Updated reference to NATO Architecture Body of Knowledge (ABoK) in chap. 2 sec. 7.5.1 (4412), added ABoK to figure 2-1 (4607) and to glossary. Added glossary entries for Measure (4198), C3 Board and ACaT (4429), Capability Configuration (4874). Added relevant publications to bibliography (4414), restructured bibliography and removed unused references (4428). Changed Architecture Kinds to Architecture Types (chap. 2 sec. 5.4), adjusted Table 2-1, and reference to TOGAF (4415, 4605). Changed Architecture Tiers to Architecture Levels in chap. 2 sec. 5.1.3 (4418). Updated Architecture Stage titles (sections 2.6, 2.7, 2.8) and corresponding figures 2-11, 2-13, 2-14 (4416) Updated figure 2-3, replaced Reference Architecture by Capability Architecture (4418). Amended definition of Architecting in chap. 1 sec. 1.1 (4604). Clarified the use of UAF DMM in NAFv4 in chap. 4 sec. 3 (4744). Editorial changes (4188, 4371, 4372, 4411, 4413, 4416, 4420, 4421, 4422, 4423, 4424, 4425, 4426, 4466, 4576, 4606, 4610, 4612, 4625, 4609, 4615, 4816, 4841, 4843, 4844, 4846, 4860, 4872, 5568).</p>
25 Sep 2020	<p>Added reference to NISP regarding versions of ArchiMate and UAF DMM required for compliance.</p>
March 2026 (4.1)	<p>Material changes to all viewpoints in Chapter 3 incorporating the NAF Information Model. Introduced Purpose and Scope for each viewpoint. Updated and merged Usage & Concern for each viewpoint Updated Representation for some viewpoints. Name changes for some viewpoints; C7: Performance Parameters => Performance Criteria C8: Planning Assumptions => Planning Constraints S6: Service Interactions => Service Sequence S8: Service I/F Parameters => Service Constraints</p>

	<p>L2: Logical scenario => Logical Structure L3: Node Interaction => Logical Interaction Lr: Lines of Development => Logical RoadMap P1: Resource Types => Resource Taxonomy P3 Resource Connectivity => Resource Interaction Pr: Configuration Management => Resource RoadMap</p> <p>Added Information Model cutouts for each viewpoint. Representation examples from each viewpoint replaced with NAFv4 compliant examples. Added Chapter 3 sections 2.2 to 2.4 detailing changes to viewpoints. Removed Chapter 1 Section 6.3.3. Fixed numerous typographical errors throughout the document and alignment to Oxford concise spelling. Improved document referencing and numbering for easier navigation.</p>
--	--