NCOIC Interoperability Framework (NIF™)

NIF™ Solution Description
Reference Manual (NSD-RM)

NIF Architecture Concepts Functional Team

Document Authors:

Gundars Osvalds – Northrop Grumman Corporation
Mark Bowler – The Boeing Company
Allen Jones – The Boeing Company
John Noble – Johns Hopkins University, Applied Physics Laboratory
John Yanosy – Rockwell Collins
Lars Schylberg – Saab
Robert Palmqvist – Saab
François-Xavier Lebas – Thales

NIF™ v2.1
NSD-RM v1.2
10 November 2010

[NSD-RM v1.2 incorporates approved updates to appendix 1.7]
Executive Summary

The Network Centric Operations Industry Consortium’s (NCOIC™) primary mission is to provide enabling guidance to accelerate the introduction of Network Centric Operations (NCO) and advance system interoperability across multiple domains of operation. Within the NCOIC process, the NCO Interoperability Framework (NIF™) provides overarching architectural guidance about the development of network centric systems. The NIF is an architectural framework that helps system architects and system engineers to design interoperable products and architectures, by supporting them with resources such as patterns, principles and methodologies. Whenever relevant, those resources are based upon standards.

As a single, monolithic framework may not address the many disciplines required to achieve systems interoperability, NIF is created as a “Framework of Frameworks” umbrella, under which resides a hierarchy of Specialized Frameworks that address capabilities such as communications, information, information assurance, semantics, services and system management. The top (root) level framework, used in developing those Specialized Frameworks, is called the Overarching Framework. Specialized Frameworks comply with the structure defined in the NIF Framework-of-Frameworks.

The use of open standards, including protocols, is critical for interoperability. The NIF selects protocols and other standards in “Patterns for Communications” (PFC)—a specific type of pattern based exclusively on protocols. NIF also addresses patterns in general and how they can be used to achieve interoperability. Patterns specify “open standards” whenever possible and provide guidance for their use. Although standards alone do foster interoperability, using only standards as guidance may preclude functionality. The framework and guidance provided by the NIF are vehicles that drive functionality.

The information provided by the NIF complements reference architectures being developed by various civil and military entities (departments, ministries, services); NIF also complements systems engineering processes and tools now used by engineering firms. Thus, NIF affords interoperability guidance that allows a firm to develop system elements/nodes that are interoperable with system elements/nodes that other firms are developing.

The service approach is a key NIF interoperability concept. NIF encourages service orientation and fosters the Service Oriented Architecture approach as a preferred pattern for information systems. At lower levels, services can provide key infrastructure capabilities (e.g. Web services). At upper levels, services support capabilities for key mission areas.
# Table of Contents

**Executive Summary** ......................................................... 2

1 Introduction ................................................................................................................. 7

1.1 Goals ................................................................................................................. 9

1.2 Scope ............................................................................................................. 11

1.3 NIF Architecture Concepts .............................................................................. 13

1.4 Contents ......................................................................................................... 17

2 The NIF Overarching Framework – OAF ................................................................. 18

2.1 Overarching Architecture Concept ................................................................. 18

2.2 Interoperability Concept ................................................................................. 19

2.2.1 Net-Centricity Maturity .................................................................................. 19

2.3 Overarching Principles ...................................................................................... 21

2.3.1 Principle Structure ......................................................................................... 21

2.3.2 Overarching Architecture Principles ............................................................ 22

2.3.3 Basic Tenets ................................................................................................... 22

2.3.4 Prerequisite Principles ................................................................................. 23

2.3.5 Architecture Principles ................................................................................ 26

2.3.6 Architecting Principles .............................................................................. 38

2.4 Overarching Patterns ....................................................................................... 47

2.4.1 Architecture Style Patterns .......................................................................... 47

2.4.2 Network Centric Enterprise Integration Pattern ............................................ 47

2.4.3 Secured Network Centric Architecture Patterns ............................................ 47

2.5 Overarching Architecture Methodology ............................................................... 48

2.5.1 Life-cycle Model ........................................................................................... 48

2.5.2 Architecting Context ..................................................................................... 51

2.5.3 Architecting Workflow .................................................................................. 51

3 Specialized Frameworks ....................................................................................... 62

3.1 Net-centric Services Framework ....................................................................... 62

3.2 Communications Framework ............................................................................. 62

3.3 Semantic Interoperability Framework ............................................................... 62

3.4 Information Assurance and Security Framework ............................................... 63

A Appendix A - NIF Model ..................................................................................... 64

A.1 FoF Model (Meta-architecture) ........................................................................... 64

A.1.1 Model Classification ...................................................................................... 64

A.1.2 Operational Descriptions (OD) .................................................................... 66

A.1.3 Principles ...................................................................................................... 74

A.1.4 Processes & Tasks ......................................................................................... 74

A.1.5 Interfaces ...................................................................................................... 75

A.1.6 Network Centric Views .............................................................................. 77

A.1.7 Network Centric Patterns .......................................................................... 83

A.2 Repository Model ............................................................................................. 100
A.2.1 Rationale ..........................................................................................................................101
A.2.2 Architecture Description .................................................................................................101
A.2.3 Architecture Document ..................................................................................................102
A.2.4 Repository Description ................................................................................................103
B ANNEX B – “Don’ts” and Anti-patterns .............................................................................105
C ANNEX C – Modeling .........................................................................................................106
C.1 Metamodeling Framework - Introduction ........................................................................106
C.2 NIF Model Goals ............................................................................................................106
C.3 Methodology Strategy ....................................................................................................106
C.4 Overview .........................................................................................................................107
C.4.1 Architecture within Enterprise Context .......................................................................107
C.4.2 Operational, System and Technical Descriptions .........................................................108
D Annex D – Technical Readiness (TRL) Scale ......................................................................121
E ANNEX E – Changes Log and Open Issues .......................................................................123
F Annex F – References ..........................................................................................................124
Table of Figures

Figure 1: NIF FoF Heirarchy .................................................................7
Figure 2: NIF Relationships ................................................................8
Figure 3: NIF Viewpoint ....................................................................9
Figure 4: Enterprise Relationships .....................................................11
Figure 5: Relationship Categories for Enterprise Systems .................12
Figure 6: The Global NCOIC Engineering Process ............................13
Figure 7: Layered Capability Model ....................................................16
Figure 8: Architecture Relationships ...............................................18
Figure 9: NIF Interoperability Model ...............................................19
Figure 10: from 1-to-1 to 1-to-many ................................................29
Figure 11: Short-term integration mode ............................................30
Figure 12: Mid-term integration mode ................................................31
Figure 13: Long-term integration mode ............................................32
Figure 14: Architecture Modeling Steps ..........................................39
Figure 15: Spiral and Evolutionary Life Cycles .................................49
Figure 16: Methodology Elements ..................................................50
Figure 17: Generic Engineering Workflow .......................................51
Figure 18: The NIF Process ...............................................................52
Figure 19: Architecting Workflow and Processes ............................54
Figure A-1: System Elements are composed of User Applications .70
Figure A-2: The Relationships Between CONOPS, Use Cases, and Systems Elements 72
Figure A-3: An Example of a Use Case Description ........................73
Figure A-4: Describing a Process/Task ..............................................75
Figure A-5: Example NOV-3 IER Matrix ..........................................81
Figure A-6: Top-down Relationship between Patterns ....................86
Figure A-7: NIF Repository Object Model .......................................103
Figure C-1: Architecture Decomposition .......................................107
Figure C-2: Life History .................................................................108
Figure C-3: Operational Description ...............................................111
Figure C-4: System Architecture Description ................................113
Figure C-5: Technical Description ..................................................115
Figure C-6: NIF Framework Relationships ....................................116
Figure C-7: RAS Core Model .........................................................118
Figure C-8: NIF/RAS Model Mapping .............................................119
Table of Tables

Table 1: NIF Stakeholder Roles ................................................................................................................................................................................................. 8
Table 2: NIF Support for Stakeholders ......................................................................................................................................................................................... 9
Table 3: Enterprise System Relationships ................................................................................................................................................................................... 12
Table 4: Network Centric Assurance Levels .............................................................................................................................................................................. 20
Table 5: NIF Process Summary ................................................................................................................................................................................................. 54
Table 6: Primary NIF Artifacts ................................................................................................................................................................................................ 64
Table A-1: Static Interface Description Elements ......................................................................................................................................................... 76
Table A-2: Dynamic Interface Description Types .............................................................................................................................................................. 76
Table A-3: Architectural View Products ..................................................................................................................................................................................... 78
Table A-4-1: Cover Page Description ....................................................................................................................................................................................... 87
Table A-4-2: Description Outline .......................................................................................................................................................................................... 88
Table A-4-3: Implementation Guidance ................................................................................................................................................................................ 93
Table A-4-4: Verification ................................................................................................................................................................................................ 96
Table C-1: OD Model Elements ................................................................................................................................................................................................ 112
Table C-2: System Architecture Description Elements .............................................................................................................................................. 113
Table C-3: Technical Description Elements ........................................................................................................................................................................... 115
Table C-4: NIF Framework Relationship Elements ....................................................................................................................................................... 117
Table C-5: Mapping Between NIF/DoDAF Views ................................................................................................................................................................. 120
Table D-1: Technical Readiness Levels .................................................................................................................................................................................... 121
Table E-1: NSD-RM Change Log and Open Issues ......................................................................................................................................................... 123
1 Introduction

The NCOIC Interoperability Framework (NIF™) provides an overall construct and specific guidance for net-centricity and interoperability. Information in the NIF complements reference architectures, systems engineering processes and tools used by developers and integrators. NIF provides interoperability guidance so that one NCOIC member firm can develop system nodes that interoperate with those being developed by another member firm. As a single monolithic framework may not address needs of the technical architecture capabilities for systems interoperability, NIF is defined as a “Framework of Frameworks” (FoF): an overall structure to allow a consistent framework description for each technical discipline.

The FoF is a tree-based hierarchy of frameworks whose top level is the Overarching Framework (OAF) and whose “children” are Specialized Frameworks (SF): these define the basic structure of a framework and the basic relationships that exist between frameworks and extensions to those relationships.

![Figure 1: NIF FoF Heirarchy](image)

For consistency between frameworks, the NIF OAF provides overarching guidance for system architecting and engineering. OAF’s guidance drives the Overarching Architecture specification and constrains Specialized Frameworks through inheritance and traceability. Where possible, guidance contained in the NIF and SF is standards-based. Although standards alone do foster interoperability, using only standards as guidance may preclude functionality. The framework and guidance provided by the NIF are vehicles that drive functionality.

FoF can be seen as a meta-architecture framework. Each framework gathers a set of resources that support architecture design for the considered framework’s discipline.

NIF’s purpose is to provide interoperability guidance to enterprise designers and the systems that compose those enterprises. The NIF also provides architectural guidance for the Overarching Architecture. The Overarching Architecture is part of the Enterprise Architectures that describes the interactions and exchange of information within the Enterprise of Enterprises.

Patterns and Specialized Frameworks—as guided by NIF’s Overarching Framework—support Enterprise Architecture designs. The patterns provide architecture guidance that enable specific network centric capabilities and mitigate specific network centric interoperability problems. A pattern is a canonical solution to a problem and, as such, is considered to be proven. In general, a pattern is not a finished design that can be applied. Rather, it is a description, template or model for solving a problem; sometimes it must be tailored and then can be reused in different situations. Patterns typically show relationships and
interactions among constituent artifacts (such as services, objects, or functions) without specifying the final application of those artifacts in the system design.

The architecture guidance rules (Overarching Architecture and Overarching Architecture Views) along with the Architecture Framework standards inform the architect in creating the Enterprise Architecture design. Enterprises are composed of systems, people, processes, organizations, information and material. The focus of this document is interoperability among the enterprise’s systems that are part of the enterprise. These relationships are shown in Figure 2.

Figure 2: NIF Relationships

NIF supports users and stakeholder roles described in Table 1 when architecting net-centricity.

<table>
<thead>
<tr>
<th>Role/Category</th>
<th>Customers</th>
<th>Suppliers</th>
<th>Certification Authority</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIF Users</td>
<td>Enterprise Architect</td>
<td>System Architect and System Engineer</td>
<td>Evaluator</td>
</tr>
<tr>
<td>Other stakeholders</td>
<td>Acquirers, Testers, Operational Users, Communities of Interest (COIs), Trainers, and Sustainers</td>
<td>System Integrator, System Verifier</td>
<td>Auditor</td>
</tr>
</tbody>
</table>

NIF support for stakeholders and program phases is summarized in Table 2:
Table 2: NIF Support for Stakeholders

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>Program Phase</th>
<th>Area supported by NIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acquirer</td>
<td>Acquisition</td>
<td>Specify acquisition requirements related to net-centricity</td>
</tr>
<tr>
<td>Enterprise Architect</td>
<td>Acquisition</td>
<td>Specify NCO environment</td>
</tr>
<tr>
<td>System Engineer</td>
<td>System Design</td>
<td>Design network centric constructs</td>
</tr>
<tr>
<td>System Integrator</td>
<td>Integration</td>
<td>Identify integration tests.</td>
</tr>
<tr>
<td>System Verifier</td>
<td>Verification</td>
<td>Identify verification tests.</td>
</tr>
<tr>
<td>Evaluator</td>
<td>Verification</td>
<td>Identify inputs to check conformance.</td>
</tr>
<tr>
<td>Auditor</td>
<td>Verification</td>
<td>Identify inputs to check conformance.</td>
</tr>
<tr>
<td>Tester</td>
<td>Validation</td>
<td>Identify validation tests.</td>
</tr>
<tr>
<td>Operational User/CoI</td>
<td>Validation</td>
<td>Ensure traceability with needs.</td>
</tr>
<tr>
<td>Trainer/Sustainer</td>
<td>Operation</td>
<td>Help building training/sustainment plan.</td>
</tr>
</tbody>
</table>

1.1 Goals

NIF’s primary goal is to help users, mainly enterprise and system architects, to identify and express network centric needs and to specify network centric architectures. This is done by creating a set of views about the enterprise from different perspectives (enterprise, business, operational, system) using the approach that a network essentially is composed of acting nodes linked by relationships. This concept is illustrated in Figure 3.

**Net-Centricity** Goal. If net-centricity could be distilled into a single word, that word might be “teaming,” since net-centricity’s goal is to allow organizations to “team.” Practically speaking, net-centricity leverages evolving networking capabilities to improve global enterprise efficiency. With this goal in mind, net-centricity properties can be summarized as:

**Environment awareness:** the ability to passively acquire useful information (sensors) and to actively seek (discovery) new information (actuators).
**Mutual sustainment:** the ability to share with other systems and to use their capabilities through common policies that support a will-to-share, thus enabling systems to team together to support their individual and joint missions.

**Autonomy:** the ability to continue operations even when partially or completely disconnected from the network for short or long periods of time.

**Trust:** supports the ability to share the right information and to assure that the information is distributed to the users approved to receive it.

More generally, net-centricity enables providing the right information to the right person at the right location, in the right time.

NIF supports these network centric properties by relying on core NCO principles such as:

- Connectivity
- Security
- Interoperability
- Discovery
- Agility
- End-to-end quality of service

These principles are discussed in greater detail later in this document. They challenge existing system and enterprise-centric approaches because—-in the network centric context—-they span existing system and enterprise boundaries. Teaming implies enlisting diverse systems, actors and organizations to achieve a common goal. Network centric teaming works because constituent systems and actors interact with each other, via the network. The challenge is how to bridge a diverse set of organizations, systems, perspectives, goals, and frames of reference so that ad-hoc teams can address real-world events.

**Architecture Goal:** before examining the components of a network centric architecture, we must define the word “architecture.” For the purposes of the NIF, an architecture is a formal description of a system; otherwise, a detailed, component-level system plan that guides implementation. According to IEEE 1471:

> “the architecture’s description details the structure of components, their interrelationships and the principles and guidelines governing their design and evolution over time.”

This concept is amplified in later chapters.

**Architecting Goal:** architecting is all about design: but all design is not architecting. Designing is the recursive application of architecting and implementing where “architecting” defines the structure and the interfaces, while “implementing” provides the functionality. This is also discussed in later chapters. ISO 15288 (Systems Engineering) states:

> “any project must implement the following in accordance with applicable organization policies and procedures, with respect to the Architectural Design Process: (1) define the architecture, (2) analyze and evaluate the architecture and (3) document and maintain the architecture.”
1.2 Scope

NIF’s scope encompasses information and communications systems whose architectures are constrained by an enterprise context and whose engineering is integrated into NCOIC’s global process context. As a result, NIF specifies an OAA that defines the way a set of enterprises systems is connected in a network centric fashion.

**Enterprise Context**. Network centric systems should not be viewed in isolation. They are tools that support individual or collective enterprises. Any solution to the challenge of network centric interoperability must support the enterprise’s needs, organization and processes.

An enterprise can be viewed as “a team of associates that assembles assets to achieve common business goals with the intent to reap the benefits of that collaboration.” Associates can be individuals or organizations that are linked by some sort of contract or compact. This definition applies to all types of organizations including for-profit, non-profit, civil and military.

Guided by business strategy and ruled by business policy, enterprise goals are reached through missions supported by capabilities. Capabilities are generally achieved by a mix of human and material means—e.g., systems and people. An enterprise is a continuously evolving system with goals that change over time and adapt to mutable environmental factors. The enterprise itself can change the environment. Changes to the internal structure of an enterprise support evolving goals and increase efficiency. An enterprise:

- acts within a generally complex, multi-dimensional environment
- collaborates with, competes with, or fights against other enterprises within its environment
- is supported by other enterprises and organizations whose intermediate goals achieve some part of the host’s goals and whose strategy and policy are guided by the host

**Enterprise relationships** - Figure 4 is a typical topological view of an enterprise of interest (E1) that points out the 3 relationships types that enterprises may have:

- aggregation
- sharing
- association

Inside the same enterprise context, the enterprise of interest (E1) aggregates another (sub) enterprise (E11), shares assets with another jointed enterprise (E2), and is associated (e.g., uses, provides, competes, fights) with another enterprise (E3). Joint

---

1 Note that NIF does not need an extensive description of the enterprise level but does need to capture the subset that constrains the enterprise systems environment.
enterprises are ruled by internal policy. They agree on a subset of common policies since their policy must conform to national or international regulations that cannot be changed. Systems that support enterprises interact directly and/or indirectly with internal or external actors (systems or roles). The relationships categories between Enterprise Systems can be identified as shown in Table 3:

**Table 3: Enterprise System Relationships**

<table>
<thead>
<tr>
<th>Internal Enterprise System to:</th>
<th>IES2EC</th>
<th>Internal Enterprise System</th>
<th>IES2IE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise Context</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal Enterprise Role</td>
<td>IES2IR</td>
<td>External Enterprise System</td>
<td>IES2EE</td>
</tr>
<tr>
<td>External Enterprise Role</td>
<td>IES2ER</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Figure 5: Relationship Categories for Enterprise Systems](image)

Depending on the category, relationships may be more or less constrained by security rules and regulations that govern an enterprise. As shown in Figure 5, roles affect internal and external enterprise actors who interact (e.g., solicit, invoke) with enterprise systems. Roles affect the access to objects within the model and the operations/services that can be obtained. Within the architecture, discretionary and mandatory controls should be used to ensure that actors are not able to have access to all the data, processes and services described in the model.

**Global NCOIC Process Context:** Inside NCOIC’s global engineering process (Figure 6) the NIF is a common framework of architecture guidance to achieve interoperability. Starting from customer requirements (CRs), scenarios and contextual information using Systems, Capabilities, Operations,
Programs, and Enterprises (SCOPE) dimensions, NIF guides the design and testing of general and domain-specific network centric system overarching architectures. As part of its output, NIF produces specifications, frameworks, and patterns that may be implemented using Building Blocks (BB) and then assessed using Network Centric Assessment Tool (NCAT). The reader is referred to the “Enterprise scope dimensions” of the NCOIC SCOPE Model for additional information on the type, size, and cohesiveness SCOPE dimensions as well as the breadth/depth and openness of enterprises.

1.3 NIF Architecture Concepts

Global NIF concepts are the overall abstractions that drive its process. Those concepts fall into four mandatory categories:

- Architecture Types
- Architecting Framework
- Overarching Framework
- Assessment

Architecture Types - NIF considers the following decomposition² of fundamental architecture types:

- Enterprise architecture is the global architecture which describes an enterprise’s environment, missions and capabilities. Enterprise architecture breaks down in segments(s). NIF recommends describing the subset that constrains embedded systems’ environment.

² This decomposition is compliant with the structure used by ISO and architectural frameworks such as DoDAF and NAF.
- Segment architecture contains the implemented subset of the enterprise vision.
- System architecture describes the collection of system components used to accomplish a specific capability and a logical representation of the system capabilities. System architecture may be network centric either by construction (greenfield systems) or by using façades (gateways, for example) for existing systems. NIF addresses the system architecture façade (that part of system architecture which is exposed to other systems through patterns).

NIF describes the OAA—the part of enterprise architecture that defines overall relationships and principles to apply to enterprise systems, so that they can be joined as a network centric group of enterprise systems. Specifying the OAA is one of NIF’s main goals.

The OAA is usually described using three views:
- Conceptual: the abstract representation of the enterprise
- Logical: the representation of the enterprise
- Instantiation: the abstraction of the logical description in the form of a physical system that supports hardware and software solutions

Specific implementation and guidance for these views can be found in architecture frameworks such as the Zachman Framework, TOGAF, DODAF, MODAF, or NAF.

Also, to serve architectural evolution over time, each exhibits a life-cycle state:
- Baseline (“as-is”)
- Interim target (stage during transition)
- Target (“to-be”)

The “service approach” correlates with NIF’s interoperability concept. NIF encourages service orientation and fosters the SOA³ approach as a preferred architecture pattern for information systems. At lower levels, services can provide key infrastructure capabilities (e.g. Web Services). At upper levels, services support key mission capabilities.

**Architecting Framework:** an architecting framework, often termed an architecture framework, is a tool that assists in producing organization-specific architectures. To achieve its goal, the architecture framework provides a set of resources that consist of artifacts. Among them are: requirements, tenets, methods and databases of component standards, specifications, products and their interrelationships. These artifacts are used to build architectures. An architecting framework can be used to develop a broad range of architectures. It describes a method for designing a system as a set of building blocks, and for showing how the building blocks fit together. It contains a set of tools and provides a common vocabulary. It also includes a list of recommended standards and compliant products that are used to

³ Refer to the NCOIC Lexicon
implement the building blocks.\footnote{From The Open Group’s TOGAF v8.1} As a framework, NIF provides context, general guidance (e.g., principles) and detailed guidance (e.g., patterns) to specify the Overarching Architecture.

Recognizing that a single framework cannot address all capabilities and disciplines involved in network centric systems, NIF provides the Framework of Frameworks (FoF) metamodel. The FoF defines the structure of network centric frameworks and the relationships between them. It also allows extending the framework’s structure to support specific domain or technical needs. The FoF defines the structure of the Overarching Framework as well as any Specialized Frameworks that may be developed separately to support architecture capabilities.

As defined by the FoF:

- Framework structure promotes the development of concepts, principles, processes and products within each framework.
- Relationships among frameworks promote the use of association, aggregation and inheritance links among frameworks.

**Overarching Framework (OAF):** In addition to defining the OAF’s structure in the FoF, the NIF also defines OAF’s contents. The OAF contents contain overall development assets that should be applied “as-is” or as further elaborated by Specialized Frameworks that support architecture capabilities. A Specialized Framework consistently refines the same concept in the OAF.

The OAF considers the following architecture capabilities:

- **Application capability:** gathers the set of domain-dependant functionalities provided by applications that support a set of operator tasks through a human system interface.
- **Community of interest (COI) capability:** gathers consistent sets of shared functionalities for domain-dependant applications. A typical example is the common operational picture that displays the current situation of enterprise assets.
- **Integration capability:** provides facilities for integrating enterprise applications together over the network, for example: discovery, publishing, composition, and services.
- **Communications capability:** defines fixed and mobile communications protocol functions
- **Information capability:** defines information and data structures from bit streams to knowledge.
- **Information assurance capability:** supports assured availability, integrity, authentication, confidentiality (including authorization), on repudiation, and defense-in-depth.
- **System and network control capability:** functionalities for managing systems and networks within a network centric environment.
These architecture capabilities can be described as a layered model, as shown in Figure 7. This layered view is a facility to determine dependencies between architecture capabilities but does not assume any specific reference model or reference architecture.

Figure 7: Layered Capability Model

Assessment: When enterprises desire to transform their business using net-centricity, changes are required at several levels and within many areas of the enterprise:

- At the enterprise level, changes will affect people, organizations and processes that drive enterprise strategic, operational and tactical vision.
- At the system level changes are likely to impact existing and planned systems.
- At the technical level changes may require new techniques and technologies.

Changes cannot be coordinated and achieved in a single step due to financial, industrial and human reasons. Multiple, incremental steps are needed: each step is driven by a maturity level that defines a consistent set of capabilities. The capabilities are supported in-depth across the enterprise, system and technical levels. Together, these maturity levels define a maturity scale.

For example, the first maturity level might be defined as:

“Communicate: enable voice and data communication among all enterprise personnel.”
- At the enterprise level, this maturity level may require people to learn to use existing or new communication devices effectively.
- At the system level, this maturity level may require better integration of existing capabilities through gateways and/or providing new communications capabilities.
- At the technical level, this maturity level may require new technology like software-defined radios.

An example of the next higher maturity level might be:

“Inform: enable any enterprise community of interest to share information in conformance with evolving and configurable security policies.”

In this example, the “inform” maturity level has a broader scope than the “communicate” maturity level because it involves additional functions and entities.
Network centric maturity cannot always be assessed using a single, complex and composite scale, but it may be assessed using several dimensions, depending on context and domain. For this purpose, the NIF recommends three complementary assessment models (1) Architecture Maturity Model, (2) Engineering Readiness Model, and (3) Network centric Assurance Model. These models are discussed in more detail in section 2.2.

1.4 Contents

This document is a reference manual that lists and describes the set of resources that NIF provides to stakeholders who want to specify and design architectures for network centric systems. The NIF has three separate deliverable volumes:

- NIF Scope and Problem Statement (NSPS) describes the NIF problem space and requirements.
- NIF Solution Description (NSD) provides solutions to NSPS and contains:
  - A reference manual (NSD-RM) that summarizes a set of general NIF guidelines and architecting rules.
  - A user’s guide (NSD-UG) that describes how to apply the NSD-RM for user profiles and use cases,
- NIF Approach and Rationale (NAR) provides rationale that justifies the design decisions that respond to NSPS requirements and lead to selection of NSD solutions.

NIF will access a shared assets catalog that contains guidelines in the form of “reusable assets” such as:

- A pattern library that contains patterns used in generating network centric solutions to problems in various domains.
- An OD library that contains a set of Operational Descriptions.

Within this Reference Manual, Section 2 provides detailed guidance on architecture and architecting. Section 3 describes the NIF Overarching Framework, including concepts, principles, patterns and the overarching methodology. Section 4 addresses Specialized Frameworks.
2 The NIF Overarching Framework – OAF

The NIF Overarching Framework (OAF) provides a set of resources that 1) provides the guidance to elaborate a network centric conformant OAA, and 2) will drive the content of the Specialized Frameworks by means of refinement, inheritance, traceability and relationships. As with all frameworks governed by the NIF, the OAF provides concepts, process, principles, and products.

2.1 Overarching Architecture Concept

The Overarching Architecture (OAA) defines the overall, high-level, long-term system-of-systems architecture that will be refined into reference architectures (RA) and then into technical architectures (TA).

In comparison with the RA or TA, the OAA captures architectural concept invariants. Figure 8 shows the relationships among OAA, RA and TA types:

Figure 8: Architecture Relationships
2.2 Interoperability Concept
The Global Network Centric Interoperability model is pictured in Figure 9.

![NIF Interoperability Model](image)

Figure 9: NIF Interoperability Model

The NCOIC only addresses the network transport and information services layers of this model. People and processes are the responsibility of the domain or enterprise. However, decisions made in the people and processes layers can impact the types of information services that systems and enterprises expose to each other and the information models and frames of reference that are used to represent those services on the network. NCOIC’s SCOPE model helps to facilitate the exploration of the possible ramifications of those decisions.

NIF recommends that stakeholders develop a business process model and an enterprise information model early in the architecture development process. Operations are described as set of interacting processes using business process modeling (BPM).

2.2.1 Net-Centricity Maturity
As discussed in section 1.3, assessing an enterprise’s network centric maturity is a vitally important aspect of the transformation process. Only by defining and implementing multiple maturity models can a full assessment of network centric maturity be made across all organizational levels. NIF recommends frequent assessments to support incremental improvements across the organization and within the systems it develops.

For this purpose, the NIF recommends three complementary assessment models described in the sections 2.2.1.1, 2.2.1.2, and 2.2.1.3. It is important to note that NIF does not provide or enforce any specific...
assessment methodology for these models, but rather provides linkages to support the methodologies that are selected.

2.2.1.1 Architecture Maturity Model (AMM) Assessment

The goal of an AMM is to assure that system and human capabilities are consistent enough to operate together efficiently. Maturity models are recognized as key enablers for network-centricity, therefore the NIF recommends implementing an AMM. When the enterprise has developed an overall Capability Maturity Model (CMM), the AMM should be used to refine the enterprise CMM, at the architecture level. The NIF does not specify a particular AMM, however the NATO NNEC levels of maturity (de-conflict / inform / collaborate / team) and the Software Engineering Institute’s Capability Maturity Model Integration (CMMI) scale (using five maturity levels) are excellent templates for developing an AMM. The enterprise decides to select or define a relevant model to apply.

2.2.1.2 Engineering Readiness Model (ERM) Assessment

The goal of the ERM is to assure the architecture has reached the desired level of technical/engineering readiness. The NCOIC Guide defines nine technical readiness levels (TRL). The NIF does not provide tools to assess engineering readiness, but does provide attribute fields--within the NIF model--where TRLs can be recorded.

2.2.1.3 Network Centric Assurance Model (NAM) Assessment

The goal of the NAM is to define network centric assurance levels (NCAL) that help to build trust in the network centric capabilities. The NCALs are similar to those contained in the common criteria⁵ and use a seven-level scale as shown in Table 4:

<table>
<thead>
<tr>
<th>Level Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCAL-1</td>
<td>NCAL1 - functionally tested</td>
</tr>
<tr>
<td>NCAL-2</td>
<td>NCAL2 - structurally tested</td>
</tr>
<tr>
<td>NCAL-3</td>
<td>NCAL3 - methodically tested and checked</td>
</tr>
<tr>
<td>NCAL-4</td>
<td>NCAL4 - methodically designed, tested and reviewed</td>
</tr>
<tr>
<td>NCAL-5</td>
<td>NCAL5 - semi-formally designed and tested</td>
</tr>
<tr>
<td>NCAL-6</td>
<td>NCAL6 - semi-formally verified design and tested</td>
</tr>
</tbody>
</table>

⁵ From the ISO/IEC 15408
Each level defines additional assurances achieved through testing and improves confidence that net-centricity will be delivered. The suggested NAM methodology is the one of Common Criteria, but applied to all net-centricity rather than just security. Using the NCAL in the Common Criteria, confidence is based upon the type and method of testing that has been used to assess functionality, starting with simple informal methods and going to more exhaustive formal methods. Typically, overall system confidence level is limited by the lowest level of its components; for example, when most system components are rated at five, but one is rated at two, the overall system rating will be two.

As with the other maturity models, the NIF does not require a particular NAM, but does suggest that one be used.

2.3 Overarching Principles
The NIF OAF contains a set of fundamental overarching principles that should be applied to any system or enterprise that strives to achieve net-centricity.

2.3.1 Principle Structure
The structure for principles contains the following elements:

<table>
<thead>
<tr>
<th>Identifier</th>
<th>A unique identifier (mandatory) constructed as:  &lt;source document&gt;&lt;framework name&gt;.”principle”. &lt;category&gt;.&lt;principle name&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A short 1-to-3 word descriptive name for the principle</td>
</tr>
<tr>
<td>Source</td>
<td>The sources or references for the principle (mandatory)</td>
</tr>
<tr>
<td>Version</td>
<td>A version number (e.g., V0.5, V1.0, V2.2, etc.) with V1.0 being the first formal release of the principle (mandatory).</td>
</tr>
<tr>
<td>Description</td>
<td>A concise free-text description of the principle and its tenets (mandatory).</td>
</tr>
<tr>
<td>Context/Preconditions</td>
<td>A free-text description of the prerequisites that must be met prior to applying this principle (mandatory).</td>
</tr>
<tr>
<td>Results/Post-conditions</td>
<td>A free-text description of the expected results and conditions that should be achieved by applying this principle (mandatory).</td>
</tr>
</tbody>
</table>
Relevant Domain
List the domain(s) for which this principle applies (for example, Sense and Respond Logistics). If the principle applies to all domains, then list “all” (optional).

Principle Relationships
List any relationships between this principle and others (optional).

2.3.2 Overarching Architecture Principles
The NIF Overarching Architecture principles guide network centric architecting and serve the following purposes:

- Provide overall framing guidance that is to be applied and refined by specialized frameworks
- Provide requirements with which framework products, especially patterns, must be consistent
- Define a basis for network centric criteria that can be used to assess the value of the architecture and alternative architectures

2.3.3 Basic Tenets

Many network centric principles contain a set of general-purpose design tenets. These tenets help users to understand the basis for achieving net-centricity and are listed below.

Enforce modularity - Modularity assists with replacement and upgrade, and often reduces coupling complexity (e.g.: interfaces). When considering methods to achieve modularity, using a component-based approach is the most advanced technique. The NIF recommends applying the service-orientation paradigm, as described in the Net-Centric Services Framework, to define the overarching architectural facilities.

When compared with other system engineering approaches, applying service orientation improves most architecture qualities, such as decoupling, agility, flexibility, and modularity.

Strictly enforce abstraction - Abstracting capabilities from the underlying technology is a fundamental information technology principle that generates more robust interfaces. Additionally, abstracting significantly eases the burden on a developer if the underlying technology changes. When applied correctly, the service-driven analysis leads to strict abstraction.

Be explicit - Many network centric problems come from implicit assumptions that are not documented or shared between networked parties. The most common implicit assumptions include:

- Assumptions about global or local context
- Assumptions about syntactic or semantic information (e.g., bits of an array)
As a general rule, every assumption must be explicit and the pre-conditions paradigm must be employed whenever possible. Reviewing the dimensions of the NCOIC SCOPE model is an excellent way to uncover implicit assumptions in these areas.

**Use (interface) standards whenever possible** - Using standards supports long-term provisions that shorten development time, assist with requirements and design changes, and decrease the total cost of ownership (TCO).

**Apply a decentralized work scheme** - Within a network of systems, decentralized schema help to keep security domains independent and foster improved availability. Decentralized schema preserve autonomy, diversity and reconfiguration; they are well suited for coalitions, federations, and joint operations.

**Preserve autonomy** - The orientation towards net-centricity emphasizes the need for autonomous behaviors in order to ensure mission continuity. This stems from the fact that the more a system uses networked resources, the more the system may be vulnerable to transient network restrictions and failures. This kind of vulnerability is common within stressful environments such as military operations.

### 2.3.4 Prerequisite Principles

Prerequisite principles are those that should be considered during the acquisition phase. The NIF OAF prerequisites principles are listed below.

#### 2.3.4.1 Architecture Maturity Model

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.amm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Architecture Maturity Model</td>
</tr>
<tr>
<td>Source</td>
<td>Adapted from SEI and NATO/NNEC</td>
</tr>
<tr>
<td>Version</td>
<td>V0.1</td>
</tr>
<tr>
<td>Description</td>
<td>Net-centricity can only be achieved through planning and effort. Network-centric elements must be tied together in an incremental manner to allow users and systems at various maturity levels to converge towards consistent network centric steps. Selecting or defining an architecture capability maturity model (AMM) is critical for achieving net-centricity. Where relevant, the enterprise should select a maturity model scale which is consistent (i.e., traceable) with those that are currently in use- (e.g., NNEC for military enterprises). The AMM model will drive the definitions of successive target architectures. Such</td>
</tr>
</tbody>
</table>
a model can later be used to consistently connect systems at the highest common level of maturity. The model will define “minimal” shared and trusted consistency points but will not include the network centric systems capability to discover additional functionality. When the enterprise has defined a Capability Maturity Model (CMM), AMM levels must be defined consistent with the CMM. There is currently no universal AMM: at this time, an AMM must be built for each network centric environment.

A good example of a four-level AMM model (inspired by the NATO NNEC maturity model) is:

1. Interchange: this level allows a team to work seamlessly in a distributed manner and perform a task from any available operational node--providing that the security policy is met.
2. De-conflict: this level ensures that concepts between parties are aligned
3. Communicate and Inform: this level provides (simple) communication and information-sharing means
4. Delegate: this level enables the delegation of some tasks between nodes

An Enterprise Capability Maturity Model (ECMM) should be defined first to ensure the AMM aligns with the ECMM.

Maturity scale that defines maturity steps.
Each step fully supports an operational goal.

### 2.3.4.2 Decouple Systems from Operations

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.Decoupling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Decoupling of Systems from Operations</td>
</tr>
<tr>
<td>Source</td>
<td>Adapted from DoD5000</td>
</tr>
<tr>
<td>Version</td>
<td>V0.1</td>
</tr>
</tbody>
</table>
| Description      | Older systems engineering approaches (e.g.: CONOPS, OCD) tend to describe operations, even at higher levels, using the systems and systems functionalities that currently support them. This leads to a tight coupling between the operations and systems, thus making changes more difficult. Enterprise operations should be abstracted from the systems that support them and then be described as systems-independent business capabilities; this facilitates dynamic changes in operations, consistent with network centric operations.
The NIF recommends the development of a business process model and an enterprise information model early in the architecture development process. Operations are described as set of interacting processes, using BPM. Current system engineering practices (e.g., CONOPS) often assume the existence of systems types when defining operations. NIF recommend using newer engineering practices that promote a capability-driven engineering that abstracts operations from supporting systems (e.g., DoD 5000 using DOTMLPF). Operations are described as set of interacting processes using techniques such as BPM.

### Context / Pre-conditions
Identify and select relevant business artifacts

### Results / Post-conditions
Business process model

### Relevant Domain
[optional]

### Principle relationships
[optional]

#### 2.3.4.3 Evolutionary Life-Cycle

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.Evolutionary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Evolutionary Life-cycle</td>
</tr>
<tr>
<td>Source</td>
<td>Adapted from DoD5000</td>
</tr>
<tr>
<td>Version</td>
<td>V0.1</td>
</tr>
<tr>
<td>Description</td>
<td>The more fluid the environment and the broader the scope, the greater the difficulty in defining requirements, the less utility in a “grand design”, and the greater the value of spiral implementations with feedback. The NIF recommends adopting an evolutionary life cycle for systems that make up the global enterprise architecture. Note that evolutionary life-cycle may also be know as spiral development</td>
</tr>
</tbody>
</table>

#### Context / Pre-conditions
Prospective risks/opportunities must be carefully managed. Maturity should be achieved in an incremental fashion using a spiral approach.

#### Results / Post-conditions
An architecture description with related risks and opportunities

#### Relevant Domain
[optional]

#### Principle relationships
[optional]
2.3.5 Architecture Principles

Architecture principles are those principles which should be applied to any architecture being developed, including those architectures defined in this document. The architecture principles recommended by the NIF OAF are listed below.

2.3.5.1 Adhere to Standards

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.standards</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Adherence to Standards</td>
</tr>
<tr>
<td>Source</td>
<td>Adapted from the Swedish Defence Material Administration FMV FMLS OAA Archprid 004</td>
</tr>
<tr>
<td>Version</td>
<td>V0.1</td>
</tr>
<tr>
<td>Description</td>
<td>System architectures should be based on open and accepted international standards, both civilian and military. Proprietary standards should only be considered when they deliver sufficient benefits for the organization without dominating the overarching or technical architectures. Standards affect the following:</td>
</tr>
</tbody>
</table>

**Operational:**
*Standards may affect internal processes and operational rules.*

**Information:**
*Information standards may affect the exchange of information in an operational process. There must be an overall model that describes the relation between different information models and their relation to operational processes.*

**Information system:**
*Information systems will be developed or selected with preference for low risk and compliance with industry standards.*

**Infrastructure:**
*Information technology will be selected with preference for low risk technologies, compliance with industry standards and possibility to integrate with current and planned overarching and technical architectures.*

**Governance:**
*Methods and tools for development of information systems shall be selected with preference for low risk and compliance with industry standards.*

**Security:**
*The standards definition process must cope with existing and emerging standards and may need to be updated. Selected standards should be in line with network centric concepts such as service orientation and service-oriented architecture.*
Standards can be in conflict with traditional systems of information classification. Security processes must be able to cope with the network centric needs for information sharing, e.g. using risk management instead of traditional non-network centric approaches such as restricted access.

<table>
<thead>
<tr>
<th>Context / Pre-conditions</th>
<th>Identify candidate standards that may solve net-centricity issues. Define selection criteria from operational requirements.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results / Post-conditions</td>
<td>Candidate standards that is relevant for the considered operational capability.</td>
</tr>
<tr>
<td>Relevant Domain</td>
<td>All</td>
</tr>
<tr>
<td>Principle relationships</td>
<td>Nif.oaf.principle.architecting.enterprise</td>
</tr>
</tbody>
</table>

### 2.3.5.2 System Interfaces Model

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.InterfacesModel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>System Interfaces Model</td>
</tr>
<tr>
<td>Source</td>
<td>Model Driven Architectures (MDA) and Model Driven Engineering (MDE)</td>
</tr>
<tr>
<td>Version</td>
<td>V0.1</td>
</tr>
<tr>
<td>Description</td>
<td>System interfaces should be explicitly defined and discoverable with timelines consistent with potential missions. Interfaces should be described using one or several interface description languages (IDL). When several IDLs are used, the mapping between IDLs shall be provided. Interfaces model impacts: <strong>Operations:</strong> The operational processes shall be able to exploit capabilities and information provided by the interfaces. <strong>Information:</strong> The knowledge of the information and the information exchanges model must be available and updated. <strong>Information system:</strong> Information system functionality must be designed to allow for changes in operations. The information system must be built (e.g., greenfield) or adapted (legacy) to implement the descriptions. <strong>Infrastructure:</strong></td>
</tr>
</tbody>
</table>

The operational processes shall be able to exploit capabilities and information provided by the interfaces. **Information:** The knowledge of the information and the information exchanges model must be available and updated. **Information system:** Information system functionality must be designed to allow for changes in operations. The information system must be built (e.g., greenfield) or adapted (legacy) to implement the descriptions. **Infrastructure:**
The infrastructure must be able to adapt to fast changes in the operational situation and support **versioning** and **time stamping**.
The infrastructure must be able to support new descriptions rapidly.
The infrastructure needs to be able to work with different technology generations simultaneously.

**Governance:**
Establishes new methods of verification and validation to ensure quality and enable support of rapid change.
Needs an organization that is able to govern rapid changes.
Establishes strict configuration control management.

**Security:**
The security model must allow securely sharing information using mechanisms such as labeling, ciphering and digital signatures

| Context / Pre-conditions | Define or select a meta-model for interface description. Select relevant IDLs
|                         | Ensure that the definition of the interface allows discovery of the interface over a network connection and implementation of the interface via a network connection without any assumption regarding the platform execution environment of the interacting systems. |
| Results / Post-conditions | An interface model is defined and consistently expressed using a common meta-model. |
| Relevant Domain [optional] | All |
| Principle relationships [optional] | Nif.oaf.principle.architecting.decoupling Nif.service-orientation.principle |

### 2.3.5.2.1 Interface Description & Management

Describing interfaces between systems (external system interfaces) is critical for interoperability and is the cornerstone of net-centricity. Interfaces shall be described using a one-to-many relationship approach, as shown in Figure 10. Interfaces shall not be described using one-to-one-relationships.
Nevertheless, the classic one-to-one IER description as recommended by several architecture frameworks (such as DoDAF) is useful for identifying data exchange needs. Interfaces are described using a static description for the structure and one or more dynamic descriptions for describing behavior.

When referring to existing information, IER descriptions may consist of a reference to that information. Static interfaces may be described using an interface definition language. Dynamic interface descriptions provide a temporal sequence of exchanges between two communicating parties. Dynamic interfaces may be described using UML sequence diagram or message sequencing charts. Interface descriptions MUST strictly apply the abstraction principle: interface definition semantics (operations and parameters) must use a syntax that is wholly independent from function/service implementation.

### 2.3.5.3 Legacy Integration

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.LegacyIntegration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Legacy Integration</td>
</tr>
<tr>
<td>Source</td>
<td>Adapted from NATO/NNEC</td>
</tr>
<tr>
<td>Version</td>
<td>V0.1</td>
</tr>
</tbody>
</table>
| Description                 | To remain compatible with the Architecture Maturity Model, NIF recommends a three-step integration strategy, as described below and illustrated in Figure 11 through Figure 13  
  - short-term,  
  - mid-term,  
  - long-term.  
  The steps shall utilize the mediation principle. |
The short-term strategy provides ad-hoc gateways between enterprises (e.g., nations in the military domain). This strategy has the advantages of being straightforward and allows easy control (e.g., security) of exchanges. This strategy can also be applied to share basic services (e.g., messaging and information sharing).

The midterm strategy provides select direct connections, in addition to ad hoc gateways between enterprises (e.g., nations in the military domain). This strategy has the advantage of being straightforward, however control (e.g., security) can be more difficult. The strategy is adapted to share limited coarse-grained services (e.g., messaging and information sharing) and to share limited applications such as the Common Operational Picture).

**Figure 11: Short-term integration mode**

The short-term strategy provides ad-hoc gateways between enterprises (e.g., nations in the military domain). This strategy has the advantages of being straightforward and allows easy control (e.g., security) of exchanges. This strategy can also be applied to share basic services (e.g., messaging and information sharing).
Figure 12: Mid-term integration mode

The long-term strategy provides direct links between the services of multiple enterprises (e.g., nations in the military domain). This strategy has the advantage of allowing a fine-grained, reconfigurable distribution, however it raises many control (e.g., security) issues. It requires a companion security approach, for example, an information assurance framework.
In all cases, prior to setup these strategies require establishing a means to handle differences in data models and protocols. Differences in data models can be processed through translation and pivot techniques; differences in protocols can be processed by using façade and wrapping techniques.

Both approaches can be used at the individual system or enterprise level. Several systems or enterprises could establish a brokered set of services that support mapping and translation among several data models, information domains, and protocols. Generally, the latter “third-party” broker approach is more scalable and economic if more than a few systems or enterprises are involved--typically, using some form of canonical representation to translate between the individual system/enterprise models. In the business world these are usually called business-to-business exchange services or information brokers.

Rationale: It’s worth pointing out that “teaming” and inter-enterprise interaction often require business model changes in order to be cost effective and scalable across a larger community. NIF discourages the notion that every system may need to be able to translate to and from every other system’s information model and scope domains.

### 2.3.5.4 Enterprise Integration Facilities

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.ncei</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enterprise Integration Facilities</td>
</tr>
<tr>
<td>Source</td>
<td>Adapted from NCES, NII, PKI</td>
</tr>
<tr>
<td>Version</td>
<td>V0.1</td>
</tr>
</tbody>
</table>
| Description      | OAA shall define a common Network Centric Enterprise Integration (NCEI) facility whose interfaces are available on each system node or proxy in order to allow dynamic enterprise systems integration. NCEI is supported by several architecture capabilities:  
- Communication  
- Information  
- Security  
- System management  
- Integration  
Those facilities typically include:  
- Negotiating syntax and Semantics  
- Discovery and subscription  
- Advertisement (Secured registry, metadata, data)  
- Transaction support  
- Security services (access control, confidentiality, integrity…) |
### 2.3.5.5 Autonomy

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.Autonomy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Autonomy</td>
</tr>
<tr>
<td>Source</td>
<td>Adapted from FMV NBD</td>
</tr>
<tr>
<td>Version</td>
<td>V0.1</td>
</tr>
<tr>
<td>Description</td>
<td>Being networked may introduce new system vulnerabilities in case of network failures. Systems shall be able to operate in offline mode.</td>
</tr>
<tr>
<td>Context / Pre-conditions</td>
<td>Identify network-related threats.</td>
</tr>
<tr>
<td>Results / Post-conditions</td>
<td>The results of applying this principle are systems that are robust and can operate ‘offline’.</td>
</tr>
<tr>
<td>Relevant Domain [optional]</td>
<td>All</td>
</tr>
<tr>
<td>Principle relationships [optional]</td>
<td>Nif.oaf.principle.architecting.Resilience</td>
</tr>
</tbody>
</table>

### 2.3.5.6 Resilience (Robustness)

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.Resilience</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Resilience (Robustness)</td>
</tr>
</tbody>
</table>
The technical architecture shall support resilience in order to minimize the effects of system failures. Failure in one node or component shall have as limited effect as possible on the overall operation. A resilient environment should enable alternate configurations.

Resilience have impact over:

**Operational:**
The organization must define the requirements for autonomous operation.

**Information:**
Data must be made available to maximize autonomy of operational units.  
The way to share data must be defined.

**Information system:**
Information systems shall be designed to enable autonomous operational units and degradation of system functionality.

**Infrastructure:**
Redundant systems may be needed.  
The infrastructure shall enable prioritization of technical resources.

**Governance:**
Processes are needed for system reconfiguration.

**Security:**
Security rules must allow for system reconfiguration.

### Context / Pre-conditions
- Identified threats and vulnerabilities.
- Target architecture.

### Results / Post-conditions
**Resilience functions identified.**

### Relevant Domain [optional]
All

### Principle relationships [optional]
Nif.oaf.principle.architecting.Autonomy

---

**2.3.5.7 Mediation (Bridging)**

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.Mediation.Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Mediation (Bridging)</td>
</tr>
<tr>
<td>Source</td>
<td>Adapted from FMV NBD “Bridging” facility.</td>
</tr>
</tbody>
</table>

---

Approved for Public Release  
Distribution Unlimited  
NCOIC-NIFv2.1NSD-RM-20101110  
NSD-RM  
Version 1.2  
Page 34 / 125
**Version**: V0.1

**Description**
The mediation principle says that whenever the internal communication architecture of an individual legacy system, safety critical system or interoperability system interfaces with the common communication architecture, there shall be a mediation facility between the two. The mediation facility shall have well-defined interfaces to both the internal communication architecture and the common communication architecture. This mediation facility serves several purposes:

- It can act as a controlled and secure data transfer mechanism.
- It can act as a data flow separator so that the internal architecture has a predictable communication environment to ensure other non-functional requirements (e.g. security, safety- and real-time requirements) are satisfied.
- It can act as a translator between different data representations or data exchange models to provide interoperability.
- It can manage Quality of Service (QoS) through pre- or post-processing. The functions may be used separately or in combination.

**Context / Pre-conditions**
Mediation requirements that usually come from existing systems shall be explicitly identified and described.

**Results / Post-conditions**
Specified mediation facility.

**Relevant Domain**
All

**Principle relationships**
Nif.oaf.principle.architecting.Interfaces
Nif.service-orientation.principle

### 2.3.5.8 Architecture Agility

Not addressed in this version.
2.3.5.9 Cooperative Governance

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.Governance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Cooperative Governance</td>
</tr>
<tr>
<td>Source</td>
<td>Adapted from FMV</td>
</tr>
<tr>
<td>Version</td>
<td>V0.1</td>
</tr>
<tr>
<td>Description</td>
<td>Overarching architectures shall promote a governance policy based on</td>
</tr>
<tr>
<td></td>
<td>• machine-readable policies</td>
</tr>
<tr>
<td></td>
<td>• distributed management</td>
</tr>
<tr>
<td></td>
<td>• collaborative management to support end-to-end QoS</td>
</tr>
<tr>
<td>Context / Pre-conditions</td>
<td>Policy objectives are defined.</td>
</tr>
<tr>
<td>Results / Post-conditions</td>
<td>Governance policy is defined</td>
</tr>
<tr>
<td>Relevant Domain</td>
<td>All</td>
</tr>
<tr>
<td>[optional]</td>
<td></td>
</tr>
<tr>
<td>Principle relationships</td>
<td>Nif.oaf.principle.architecting.acmm</td>
</tr>
<tr>
<td>[optional]</td>
<td>Nif.oaf.principle.architecting.mediation</td>
</tr>
</tbody>
</table>

Immediately following their development, enterprise systems begin to evolve. Driven by a network centric maturity model, governance is required to ensure convergence towards consistent global enterprise goals. This governance applies to all enterprise resources and drives their evolution. Regarding system resources, governance is supported by system/ network control and management functions and tools.

Governance policies are the part of the business and IT policies that control how services and systems behave and where they are instantiated. Much of the flexibility requirement is accomplished by using machine-readable policies as the means to control the system. A distributed management concept is used to manage the system from both technical and organization views. The management system shall be divided in smaller parts that are separated by poor bandwidth or completely disconnected from the main backbone system.

The system management processes and functions are used through the entire life cycle system or system parts (mission specific) from pre-deployment (development, certification, accreditation, test, etc.) through deployment, operation to un-deployment/ re-deployment.
Governance can be facilitated by orchestration as a means to deliver additional management capabilities through reuse of existing services. Although governance is often viewed from top-down management oversight and centralized planning perspectives, incentive models and mechanisms should also be considered as part of a comprehensive governance strategy--especially when other institutional context and scope factors limit the efficacy of the central planning and enforcement approach. Business model changes can motivate individual systems and enterprises to provide services desired by others and which interoperate smoothly over the network, without a lot of top-down control or enforcement. Such changes can also foster greater collaboration in the governance process.

### 2.3.5.10 Quality of Service

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.QoS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>Source</td>
<td>ISO RM-ODP (Reference Model – Open Distributed Processing )</td>
</tr>
<tr>
<td>Version</td>
<td>V0.1</td>
</tr>
</tbody>
</table>
| Description | QoS is defined at several levels of service, including the transmission of traffic over a network, parameters that address loading, providing users the correct information, the connection among providers and users (i.e., end-to-end). Higher level capabilities that must be considered when defining a QoS policy include the following:  
  - Monitor and measure service health/performance.  
  - Report and visualize service performance metrics.  
  - Monitor and enforce service level agreement (SLA) compliance.  
  - Manage service lifecycle.  
  - Log and audit service activities.  
  - Anticipate service problems and send alert notifications.  
  - Pinpoint the root cause of service problems |
| Context / Pre-conditions | Assess QoS objectives |
| Results / Post-conditions | QoS policy defined |
| Relevant Domain [optional] | All |
| Principle relationships [optional] | |
2.3.5.11  Real-time
Not addressed in this version.

2.3.5.12  Safety
Not addressed in this version.

2.3.6  Architecting Principles
Architecting principles are those principles which should be applied to the process of developing architectures. The architecting principles recommended by the NIF OAF are listed below.

2.3.6.1 Model-Driven Architecting

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.ModelDrivenArchitecting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Model Driven Architecting</td>
</tr>
<tr>
<td>Source</td>
<td>Model Driven Architectures (MDA) and Model Driven Engineering (MDE)</td>
</tr>
<tr>
<td>Version</td>
<td>V0.1</td>
</tr>
<tr>
<td>Description</td>
<td>Current engineering practices often involve (proprietary) technologies when defining systems (interfaces). This fact links system and technology changes, forcing one to change with the other (e.g., technology obsolescence also forces a system change) and makes development more difficult. To allow for easier transformation and agility, enterprise systems must be abstracted from technologies that support them (e.g., by using web-services) and a technology-agnostic architectural model shall be designed first. Then this “platform independent model” (PIM) can be translated into a “platform specific model” (PSM) using a specific set of technologies. Additionally, the PIM will be usable as a starting point for simulation, and the PSM could be used for implementation.</td>
</tr>
<tr>
<td>Context / Pre-conditions</td>
<td>Select tools, techniques, and methodologies for modeling. Educate personnel on the use of the tool, techniques, and methodology</td>
</tr>
<tr>
<td>Results / Post-conditions</td>
<td>Provide PIM and PSM architecture models</td>
</tr>
<tr>
<td>Relevant Domain [optional]</td>
<td></td>
</tr>
<tr>
<td>Principle relationships [optional]</td>
<td></td>
</tr>
</tbody>
</table>
NIF recommends the development of the OAA from abstract to concrete, combining top-down, middle-out and bottom-up approaches:

- When one or several reference models are available and relevant, the first top-down step is to use that reference model(s) as a basis to design reference architecture(s).
- Reference patterns can then be reused to complement reference architecture(s) in a middle-out manner.
- Finally, the reference architecture, as captured in the PIM, is translated into the target architecture, as captured in the PSM, using a set of technologies.

NIF also recommends the abstraction of systems from technologies by defining and implementing service-oriented front-ends. For “greenfield” (i.e., completely new systems) a component-based development approach can also help to achieve this abstraction. This approach is outlined in Figure 14:

![Figure 14: Architecture Modeling Steps](image)

### 2.3.6.2 State Enterprise Context and Missions

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Define Enterprise Context and Mission</td>
</tr>
<tr>
<td>Source</td>
<td>Adapted from BPMN as maintained by OMG (<a href="http://www.omg.org">www.omg.org</a>)</td>
</tr>
<tr>
<td>Version</td>
<td>V0.1</td>
</tr>
<tr>
<td>Description</td>
<td>Traditional systems engineering integrates technical and business considerations, while enterprise systems engineering must also integrate political, organizational and economic considerations.</td>
</tr>
</tbody>
</table>
The NIF recommends stating enterprise context and missions for both “as-is” and “to-be” states:

- State business environment: enablers and disablers, risks and opportunities
- State business goals, strategy and policy
- State business scenario, use cases, and requirements
- State business capabilities and information
- State acquisition plan for capabilities
- State security goals and security plan.
- State foreseeable business evolutions.

Use of the NCOIC SCOPE tool is recommended to characterize the network centric aspects of the enterprise context.

<table>
<thead>
<tr>
<th>Context / Pre-conditions</th>
<th>Gather as much information as allowed and possible.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Results / Post-conditions</td>
<td>Share understanding about enterprise as-is and to-be.</td>
</tr>
<tr>
<td>Relevant Domain [optional]</td>
<td>All</td>
</tr>
<tr>
<td>Principle relationships</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Usually, an enterprise model includes the following set of views:

- A business reference model (BRM) – conceptual view that provides business goals, strategy, policy, assets, operations, tasks and procedures.
- An application reference model (ARM) – logical view that provides applications (functions) that support the business model
- A technical reference model (TRM) – instantiation view that provides the technical hardware and software assets that make up the enterprise system-of-systems
- An information reference model (IRM) – all views that provides information (including semantics) that supports the business model

The NIF does not require any format for describing the information above. It is worth noting that this principle can be at least partially supported by using MoDAF or NAF acquisition and capabilities views.

### 2.3.6.3 Overarching Architecture Views

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.views</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Overarching Architecture Views</td>
</tr>
<tr>
<td>Source</td>
<td>Adapted from DoDAF</td>
</tr>
</tbody>
</table>
The NIF recommends documenting each system using an appropriate number of “views,” as appropriate to present the network centric point-of-view. Typical views are operational, system and technical:
- The operational view (OV) focuses on operational business concerns
- The system view (SV) focuses on functional/logical architecture concerns (e.g., a model-driven architecture approach using the PIM)
- The technical view (TV) focuses on implementation (e.g., a model-driven architecture approach using the PSM).

Each view shall be described using the network-driven paradigm based on the nodes/edge concepts. Also, while the SV should be derived from the OV, if possible, and the TV should be derived from the SV, both middle out and bottom approaches to architecting may entail SVs and TVs that transcend any particular OV. Systems may support multiple operational architectures and TVs may be dictated by enterprise or multi-enterprise standards.

The selected system of views should relate the following:
- Contextual: the forces that influence the value and behavior of the enterprise
- Conceptual: the abstract representation of the enterprise
- Logical: the representation of the enterprise
- Instantiation: the abstraction of the logical description in the form of a physical system that support hardware and software solutions

While security concerns can be described using the standardized views above, the NIF recommends documenting a separate view dedicated to security.

<table>
<thead>
<tr>
<th>Context / Pre-conditions</th>
<th>Results / Post-conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>State enterprise context and missions. Developing one or several scenarios and use cases is required prior to building these products.</td>
<td>Building in whole or in part, the following core set of products is considered as essential for net-centricity:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCAV-2</td>
<td>Integrated Data Dictionary</td>
</tr>
<tr>
<td>NCOV-2</td>
<td>Operational Connectivity Description</td>
</tr>
<tr>
<td>NCOV-3</td>
<td>Operational Information Exchange Matrix</td>
</tr>
<tr>
<td>NCOV-7</td>
<td>Logical Data Model</td>
</tr>
<tr>
<td>NCSV-1</td>
<td>Systems Interface Description</td>
</tr>
<tr>
<td>NCSV-2</td>
<td>Systems Communication Description</td>
</tr>
<tr>
<td>NCSV-3</td>
<td>NCSV-3 System-to-Systems Matrix</td>
</tr>
<tr>
<td>NCSV-4</td>
<td>Systems Functionality Description</td>
</tr>
<tr>
<td>NCSV-6</td>
<td>System Data Exchange Matrix</td>
</tr>
<tr>
<td>NCSV-7</td>
<td>Systems Performance Parameters Matrix</td>
</tr>
</tbody>
</table>
Each of these views is described further below as products created to document the views:

- **NCAV-2** Integrated Data Dictionary
- **NCOV-2** Operational nodes, connectivity & information exchange between nodes
- **NCOV-3** Information exchanged between nodes and the relevant attributes of that exchange
- **NCOV-7** Documentation of the system data requirements and structural business process rules of the Operational View
- **NCSV-1** Identification of systems nodes, systems, and system items and their interconnections, within and between nodes
- **NCSV-2** Identification of systems nodes, systems, and system items and their interconnections, within and between nodes
- **NCSV-3** Relationships among systems in a given architecture; can be designed to show relationships of interest; e.g., system-type interfaces, planned versus existing interfaces, etc.
- **NCSV-4** Functions performed by systems and the system data flows among system functions
- **NCSV-6** The Systems Data Exchange Matrix specifies the characteristics of the system data exchanged between systems.
- **NCSV-7** The product specifies the quality characteristics of systems, system hardware/software items, their interfaces (system data carried by the interface as well as communications link details that implement the interface), and their functions.
- **NCSV-8** Planned incremental steps toward migrating a suite of systems to a more efficient suite, or toward evolving a current system to a future implementation
- **NCSV-9** Emerging technologies and software/hardware products that are expected to be available in a given set of timeframes, and that will affect future development of the architecture
- **NCSV-11** Physical implementation of the Logical Data Model entities; e.g., message formats, file structures, physical schema
- **NCTV-1** Listing of standards that apply to systems view elements in a given architecture
- **NCTV-2** Description of emerging standards and potential impact on current Systems View elements, within a set of timeframes
This set of products can be directly mapped onto NAF but can also be mapped onto architecture frameworks such as DoDAF (US), MoDAF (UK), DAF (Australia), etc., and can also be produced using other architecting methodologies such as TOGAF (OpenGroup) or AGATE (FR).

### 2.3.6.4 Service Orientation

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.ServiceOrientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Service Orientation</td>
</tr>
<tr>
<td>Source</td>
<td>Adapted from FMV FMLS OAA Archprid 001</td>
</tr>
<tr>
<td>Version</td>
<td>V0.1</td>
</tr>
<tr>
<td>Description</td>
<td>The NIF recommends applying the service-orientation paradigm as described in the Net-centric Service Framework to define the overarching architectural facilities. When compared with other system engineering approaches, applying service orientation improves most qualities of architecture, such as decoupling, agility, flexibility, modularity, etc.</td>
</tr>
<tr>
<td>Context / Pre-conditions</td>
<td>Service orientation is quite different from traditional functional system engineering analysis. A service-oriented methodology should be selected or defined first, before analysis. Also, not all teams and disciplines are prepared to apply service orientation, so applying service orientation to a specific overarching architecture capability should be optional.</td>
</tr>
<tr>
<td>Results / Post-conditions</td>
<td>Service-oriented architecture model.</td>
</tr>
<tr>
<td>Relevant Domain [optional]</td>
<td>All</td>
</tr>
<tr>
<td>Principle relationships [optional]</td>
<td>Nif.oaf.principle.architecting.decoupling</td>
</tr>
<tr>
<td></td>
<td>Nif.service-orientation.principle</td>
</tr>
</tbody>
</table>

It is a common practice to define operational needs or requirements that lead to enterprise and system assets. This practice leads to interdependencies that can complicate subsequent operational or technical changes. This is why it is critical to abstract systems from operations and technologies using abstraction layers and concepts. The practice of using abstraction layers has commonly and successfully been applied for several decades within the IT industry. The industry has used abstraction layers to document the Architecture levels:

- Enterprise
- Segment
- System
Each of these can then be decomposed into views:
- Contextual
- Conceptual
- Logical
- Instantiation

Which are further decomposed into viewpoints of the views:
- Domain
- Business
- System
- Physical

Until recently, this abstraction principle was applied using various and often-incompatible methods. Today, the service-oriented approach provides a uniform way of abstracting layers using service-related concepts such as service interfaces, service interoperability points (SIOP), service level agreements (SLA) and quality of service (QoS).

Services fit the conceptual view of the abstraction layers defined above. Services design can then be defined in the system viewpoint.

The NCOIC plans to release the Net-Centric Service Framework in the future. This document will provide additional details about the service concept and its application to support this principle.

### 2.3.6.5 Manage Architecture Forecasts

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.forecasts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Manage Architecture Forecasts</td>
</tr>
<tr>
<td>Source</td>
<td>Adapted from IEEE 1470 &amp; DoDAF</td>
</tr>
<tr>
<td>Version</td>
<td>V0.1</td>
</tr>
</tbody>
</table>
| Description      | NIF recommend to set a focus about forecasts in the following areas:  
|                  | - Operational,                          |
|                  | - System Architectures,                 |
|                  | - Technical.                           |
|                  | Forecasts have to be continuously updated. Risks and opportunities have to be updated in accordance. |
| Context /        | State and manage forecast regarding enterprise context and missions. |
| Pre-conditions   |                                         |
### 2.3.6.6 Configuration Management

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.ConfigurationManagement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Configuration Management</td>
</tr>
<tr>
<td>Source</td>
<td>Adapted from Engineering Practices (ex: INCOSE)</td>
</tr>
<tr>
<td>Version</td>
<td>V0.1</td>
</tr>
<tr>
<td>Description</td>
<td>Traditional system engineering uses a static configuration management schema that supports pre-planned configurations. Static configuration management is focused on the development process for capabilities. To support net-centricity, NIF recommends implementing a dynamic configuration management schema that also allows for unplanned configurations and dynamic response at run-time. This schema shall provide support for multiple versioning and compatibility of interfaces.</td>
</tr>
<tr>
<td>Context / Pre-conditions</td>
<td>Identify dynamic configuration management needs. Dynamic configuration management is available at run-time and supports efforts such as compatibility verification for connecting to services.</td>
</tr>
<tr>
<td>Results / Post-conditions</td>
<td>Define a configuration management policy for net-centricity.</td>
</tr>
</tbody>
</table>

### 2.3.6.7 Manage Architecture Risks and Opportunities

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Nif.oaf.principle.architecting.risks-opportunities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Management of Architecture Risks and Opportunities</td>
</tr>
</tbody>
</table>

Results / Post-conditions
- Forecasts synthesis
- Related risks and opportunities.

Relevant Domain [optional]
- All

Principle relationships [optional]
- Nif.oaf.principle.architecting.enterprise
Traditional systems engineering focuses on managing execution risk; enterprise systems engineering manages both risk and opportunity. NIF recommends stating and maintaining risks and opportunities for enterprise architecture.

<table>
<thead>
<tr>
<th>Source</th>
<th>Adapted from DISR-on-line</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>V0.1</td>
</tr>
<tr>
<td>Description</td>
<td>Traditional systems engineering focuses on managing execution risk; enterprise systems engineering manages both risk and opportunity. NIF recommends stating and maintaining risks and opportunities for enterprise architecture.</td>
</tr>
<tr>
<td>Context / Pre-conditions</td>
<td>Consider forecasts regarding Enterprise context and missions.</td>
</tr>
<tr>
<td>Results / Post-conditions</td>
<td>Updated Risks and Opportunities.</td>
</tr>
<tr>
<td>Relevant Domain [optional]</td>
<td>All</td>
</tr>
<tr>
<td>Principle relationships [optional]</td>
<td>Nif.oaf.principle.architecting.enterprise, Nif.oaf.principle.architecting.forecasts</td>
</tr>
</tbody>
</table>
2.4 Overarching Patterns

This section lists several architecture patterns that the NIF recommends for building network centric OAA. When a pattern shapes a whole architecture, it is commonly described as an “architecture style”.

2.4.1 Architecture Style Patterns

An architecture style is an abstract approach for understanding significant relationships among the entities of some environment, and for the development of consistent standards or specifications supporting that environment. An architecture style is based on a small number of unifying concepts. It is not directly tied to any standard, technology or other concrete implementation details, but it does seek to provide common semantics that can be used unambiguously across and between different implementations.

Two examples of architectural style patterns are the Service Oriented Architecture (SOA) and the Service Component Architecture (SCA).

2.4.1.1 Service Oriented Architecture (SOA)

The SOA approach is a recommended for building information technology systems. An SOA is an architecture that results from applying SO concepts, principles, processes, and patterns. The recommended network centric approach for SOAs will be described by the “Net Centric Service Framework,” a specialized framework that will be developed by the NCOIC.

2.4.1.2 Component-based Architecture

Component-based architectures are those based on a standardized component model, such as the one defined by OMG/CCM. In general, this pattern is not applicable to existing systems since they would require significant reengineering. A component-based architecture can be used independently, but using it in conjunction with SOA is recommended for new systems.

2.4.2 Network Centric Enterprise Integration Pattern

Enterprise integration requires a set of “core” foundation services that form the basis of the entire network centric architecture. These core services define how services can be found over the network, invoked by other services and composed or synchronized. The NIF terms these services as Network Centric Enterprise Integration (NCEI) services. An example of such a pattern is the NATO NII.

2.4.3 Secured Network Centric Architecture Patterns

Taking advantage of the network--while avoiding network-related vulnerabilities--is one of the primary, and most difficult, network centric challenges. Information assurance provides a number of security patterns.
For example, the Core Network Access pattern is a security pattern under development by the NCOIC. It supports basic security properties (authentication, confidentiality, integrity) and can be extended to counter other threats, such as denial-of-service, seamlessly. Countermeasures are used to thwart denial-of-service attacks. First, one needs to recognize that there is an attack. Second, one needs to know what type of attack is being executed. Finally, one needs techniques that can adapt to counter different types of attacks that exist, or may be developed in the future. Authentication and integrity of received data packets can determine that there is an attack: those packets can be dropped, in order to avoid overloading the receiving system.

### 2.5 Overarching Architecture Methodology

#### 2.5.1 Life-cycle Model

In order to support continuous evolution of user requirements and capabilities, OAF methodology promotes an evolutionary process. It is worth noting that this Evolutionary process is a refinement of the spiral development process in the usual sense.

- Spiral development applies to system development and implies that a finished system results from the final spiral
- Evolutionary process assumes continuous evolution of systems capability and technology insertion – forever.

Other usual life cycles (V, W, Incremental) can also be used with less efficiency when development teams are not mature enough to be driven by a spiral or, more, by an evolutionary life cycle.

Spiral and evolutionary lifecycles are illustrated by Figure 15. The main difference is that the spiral life cycle is driven by a target, while the evolutionary life cycle is driven by environmental changes.
2.5.1.1 Architecting Strategy

OAF Architecting Strategy is:

• Combine top-down, middle-out and bottom-up:
  – Top-down proceeds by successive refinements starting with enterprise architecture and working toward physical architecture,
  – Middle-out introduces global architecture patterns that govern the entire architecture,
  – Bottom-up introduces specialized patterns to support middle-out patterns.
• Use a model-driven approach similar to OMG MDA and supply CIM, PIM and PSM models.
• Do not define a “silver-bullet” turnkey methodology, but rather provide methodology fragments that can be integrated within multiple engineering contexts and show usage through examples.
• Do not enforce the use of a specific architectural framework, but recommend existing (sometimes modified) products (e.g., a modified OV-3 with SCOPE dimensions and modified SV-3 with NIF Global Attributes.)
• Support two approaches to analysis
  – Function driven (classical scheme)
  – Service driven (new scheme)

2.5.1.2 Methodology Elements
Rather than defining a monolithic, continuous methodology that could be difficult to integrate with existing practices and engineering contexts, OAF promotes the concept of methodology elements (MEs) whose well-defined interfaces can be used independently and consistently, during the various steps of the evolutionary process.

Methodology is described for each one of the individual frameworks in terms of the elements shown in Figure 16 where:
- Input describes information that is required prior to flow through the ME
  o Example: operational scenarios
- Output describes information that is produced
  o Example: design artifacts
- Resources describe existing background assets that are reused
  o Example: A methodology manual like the NIF NSD-RM document itself,
- Controls are the entities that control the process
  o A control is often driven by a NIF user, assisted by another ME outputs.

The overall method, and each level of meaningful decomposition, share the description shown in Figure 16. For example, at the top level, customer and operational environments serve as inputs for several processes. The composition of the rectangular box (in the center of the figure) is often defined by the systems engineering process; if so, it can be characterized easily in the familiar terms of requirements analysis, functional analysis and allocation: followed by synthesis or design. This process should also be followed for specialized frameworks that are derived from this document; however, some tailoring will be required for specialized frameworks.
2.5.2 Architecting Context

The NIF OAF assumes the generic engineering workflow shown in Figure 17:

![Figure 17: Generic Engineering Workflow](image)

Inside this general workflow, Architecting belongs to the System Definition phase in which it is preceded by the Requirements Statement phase and followed by the Implementation phase.

2.5.3 Architecting Workflow

Architecting workflow with NIF takes place early during the system definition phase:
- It starts with operational context and requirements capture,
- It ends with delivery of a set of general (e.g.: Principles) and detailed (e.g.: patterns) guidance for defining Overarching Architecture.

The NIF process is summarized in Figure 18:
As input, the NIF expects the following to emerge from previous engineering steps:

- Customer requirements\(^6\) including:
  - functional and non-functional network centric requirements (WFR)
  - network centric scenarios (WFS)
  - network centric performance metrics (Net-Ready Key Performance Parameters (NR-KPPs))
- A set of SCOPE\(^7\) dimensions to be considered
- Catalog(s) of network centric standards
- Operational, system and technology forecasts
- In addition, NIF optionally expects an enterprise Capability Maturity Model (CMM) which will serve as a basis for establishing an architecture maturity model.

As output, the NIF produces an Overarching Architecture specification including:

- A set of OAA principles and criteria

---

\(^6\) Depending on engineering context, customer requirements can be expressed using many formats but Well Formed Requirements (WFR) format as defined by the NCOIC Requirements Validation Team is recommended

\(^7\) SCOPE is an NCOIC Product
A set of reusable assets including interfaces, services, patterns, PFCs and standard profiles built in cooperation with Specialized Frameworks

Assessment criteria derived from principles above

An Architecture Maturity Model (AMM),

A set of risks and opportunities.

Overarching principles and criteria are network centric architecture requirements that will provide the basis for selecting reusable components, standards and patterns, and to later assess net-centricity through the use of the NCAT\(^8\). The Architecture Maturity Model helps a NIF user with assessing and planning. Risks and opportunities help a NIF user to select the best trade-offs during planning. And finally, reusable assets help a NIF user to design network centric architectures.

As resources, the NIF provides:

- Two documents that support the engineering process from requirements capture through assessment.
  - The NIF solution description - Reference Manual (NSD-RM),
- NIF Reusable Assets Catalog that contains previously implemented artifacts.

Using input and resources, the NIF User produces the outputs using the architecting workflow and processes shown in Figure 19:

---

\(^8\) NCAT is an NCOIC Product
Table 5 summarizes the processes described above:

**Table 5: NIF Process Summary**

<table>
<thead>
<tr>
<th>Identifier Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>P-EVC NC</strong></td>
<td>Environment Capture</td>
</tr>
<tr>
<td><strong>P-REQ NC</strong></td>
<td>Requirements Capture</td>
</tr>
<tr>
<td><strong>P-ACS NC</strong></td>
<td>Architecture Candidate</td>
</tr>
</tbody>
</table>
| Architecture Solutions | Define architectural selection criteria  
| | Select relevant patterns using criteria  
| **P-AD NC** Architecture Design | Define alternatives architectures using patterns assemblies  
| | Define interfaces  
| | Elaborate criteria for assessment  
| **P-ROM** NC Risks & Opportunities | State net centric risks and opportunities  
| **P-FM NC** Forecasts Management | State operational, system and technical forecasts.  
| **P-CM NC** Configuration Management | Define network centric dynamic configuration policy.  
| **P-GP General** Purpose Engineering Process | Set of general-purpose engineering processes.  
| **P-GP-GA** Gap Analysis | Support gap identification, description and comparison.  
| **P-GP-IA** Impact Analysis | Support change assessment.  
| **P-GP-TA** Tradeoff (AoA) Analysis | Support analysis of alternatives.  
| **P-GP-DM** Decision Making | Support decision description, assessment and selection.  

### 2.5.3.1 P-EVC: Environment Capture

#### 2.5.3.1.1 Purpose
The role of this process is to state enterprise (of enterprises) context and constraints from a network centric point-of-view.

#### 2.5.3.1.2 Description
This process shall use the SCOPE dimensions model to characterize the context and make it more explicit.

This process may purposefully use a business process modeling approach to describe the processes that enterprises provide. The Business Process Model will likely drive the identification and definition of network centric capabilities in the next P REQ process.
NOTE: applying SCOPE will likely “enlarge” the initial architecture scope and increase architecture constraints: this is the price to pay for making systems more open and finally more network centric.

2.5.3.1.3 **Input**
Enterprise environment, missions and requirements

2.5.3.1.4 **Output**
Structured model describing:
- Enterprise goals and capabilities
- Enterprise environment including threats.

2.5.3.1.5 **Control**
Enterprise Architect

2.5.3.1.6 **Resources**
The following resources can be used:
- Business Process Model Notation
- Enterprise-related dimensions in the SCOPE model
- P-CM
- P-GP

2.5.3.2 **P-REQ: Requirements Capture**

2.5.3.2.1 **Purpose**
Extract, capture and analyze network centric architectural requirements

2.5.3.2.2 **Description**
Extract, capture and analyze network centric operational requirements. Special attention should be paid to identifying:
- operational pattern
- security objectives

Extracting network centric requirements from general requirements is not straightforward and requires prior analysis. To better elicit requirements, NIF recommends adopting a capability-based approach, expressing requirements regarding the capabilities.

Network centric requirements elicitation analysis starts by identifying exchanges required between an enterprise’s operational nodes. Here, care must be taken to abstract the one-to-one relationships that may exist between pair of nodes. Instead, exchanges must be generalized using a one-to-many relationship schema between a client node and a set of network services.
Network centric requirements may be captured using standardized architecture framework views as defined in paragraph 2.3.6.3. After elicitation, NC requirements have to be analyzed for consistency.

Identifying operational patterns can significantly help in later identifying and/or defining design patterns.

State security objectives will later frame relevant security counter measures.

2.5.3.2.3 Input
- Enterprise environment including threats
- Enterprise capability requirements including
  - Enterprise operational use cases
  - Enterprise operational scenarios
- Enterprise not-functional requirements including
  - Performances requirements

2.5.3.2.4 Output

Main output is the Network Centric Operation Description (NCOD or OD in short) which contains Network Centric operational requirements including:
- NC functions or services along with their required QoS,
- Security plan.

Other outputs are:
- Recommended NCOV views describing exchanges and information model
- Capability Maturity Model

2.5.3.2.5 Control
Enterprise or System Architect

2.5.3.2.6 Resources
The following resources can be used:
- Frameworks principles and tenets as stated in NSD-RM and MNE
- Selected operational views from architecture frameworks
- P-CM
- P-GP

2.5.3.3 P-SCS: State Candidates Solutions

2.5.3.3.1 Purpose
The process goal is to identify and develop candidate overarching architecture solutions that could partially fit into the global architecture. Special attention should be paid to existing or candidate architecture patterns: candidate architecture patterns can be found when comparing existing systems.

2.5.3.3.2 Description
In order to select solutions candidates, criteria must be elicited from architecture requirements. Special attention should be paid to performance and scope assumption criteria.

The selection process shall be staged, with each stage allowing increased relevance and a decreased number of candidates.

2.5.3.3.3 Input
Architecture requirements expressed by the OD.

2.5.3.3.4 Output
Candidate solutions including:
- Architecture Maturity Model
- Recommended NCSV views
- Related risks and opportunities
- Related forecasts

2.5.3.3.5 Control
Enterprise Architect

2.5.3.3.6 Resources
The following resources can be used:
SCOPE model
P-ROM
P-FM
P-CM
P-GP

2.5.3.4 P-AD: Architecture Design

2.5.3.4.1 Purpose
The goal is to define an architecture layout that enables multiple configurations of existing, planned and (in some way) unplanned systems to cooperate in a network centric approach. This commonly requires selecting or developing a Network Centric Enterprise Integration façade that wraps systems and enables interoperability.
2.5.3.4.2 **Description**
The general design strategy is to use a model-driven architecture approach whose goal is to devise the final architecture by successively elaborating a set of models from the most abstract, to the most concrete:
- Reference architecture model
- Conceptual architecture
- Logical architecture
- Physical architecture
The model will emphasize descriptions of the interfaces.

2.5.3.4.3 **Input**
Architecture style (style being an overarching pattern).
Candidate architectural solutions including architecture patterns

2.5.3.4.4 **Output**
Architecture document including an architecture description.
Record of architecture decisions.

2.5.3.4.5 **Control**
Enterprise Architect

2.5.3.4.6 **Resources**
The following resources can be used:
- Architecture Framework
- Interface description language
- Data model derived from the information model
- P-ROM
- P-FM
- P-CM
- P-GP

2.5.3.5 **P-FM: Forecast management**

2.5.3.5.1 **Purpose**
The goal of forecast management is to prepare for future changes and anticipate planned milestones.

2.5.3.5.2 **Description**
NIF does not enforce any specific forecasts management scheme but rather formulates which principles are to be applied for managing forecasts within a network centric environment.
Even though a network centric system may not be prepared for all future changes or context shifts, anticipating a range of potential changes is likely to result in an architecture more capable of handling unexpected changes.

NIF recommends considering forecasts at the operational, system and technical levels:

- Operational forecasts should anticipate new missions
- For new missions, system forecasts should identify which architectural changes could be required and how they could fit together
- Technical forecasts should identify technical and technology changes that may impact future systems and establish related roadmaps

SCOPE dimensions can be a useful guide for thinking about potential future operational, system and technical changes that the architecture may need to handle.

### 2.5.3.5.3 Input
Operational, system and technical descriptions.

### 2.5.3.5.4 Output
Roadmaps.
Planning.

### 2.5.3.5.5 Control
Enterprise Architect

### 2.5.3.5.6 Resources
Planning model such as Program Evaluation and Review Technique (PERT) or Gantt.

### 2.5.3.6 P-ROM: Risk and Opportunities Management (ROM)

#### 2.5.3.6.1 Purpose
The main goal of risks and opportunities management is to support architecture tradeoffs.

Within a network centric context that raises uncertainty levels, the role of ROM becomes even more critical.

#### 2.5.3.6.2 Description
NIF does not enforce any specific ROM scheme but rather formulates which principles should be applied for managing risks and opportunities within a network centric environment.

Since each architecture is a tradeoff between risks and opportunities, defining an architecture without managing risks and opportunities would be nonsense. NIF recommends managing risks and opportunities in the following ways:
Managing risks means identifying risks, assessing them (duration, impact) and taking preventive and corrective actions,

Managing opportunities means identifying opportunities, assessing them (duration, impact) and planning for their enforcement.

2.5.3.6.3 Input
Risks and Opportunities templates
Design Decisions

2.5.3.6.4 Output
Risks and Opportunities.

2.5.3.6.5 Control
Enterprise Architect

2.5.3.6.6 Resources
Risks and Opportunities model.
3 Specialized Frameworks

Specialized frameworks, as introduced in section 1.6.2 above, are an important element of the overall NIF construct. Specialized frameworks are “frameworks” as defined by the NIF, and as such, must contain all the associated resources—concepts, principles and processes. Specialized frameworks also play an important role by extending the guidance provided by the NIF OAF to capture specific guidance for the technical domain they address. This guidance must be consistent (e.g., traceable) with the guidance provided by NIF OAF resources. Each specialized framework will target a technical domain. The NCOIC Specialized Frameworks functional team is responsible for identifying, drafting and reviewing specialized framework candidates. While these frameworks must comply with the NIF Framework-of-Frameworks guidelines and must be compatible with the NIF OAF, they will be maintained as separate documents from the NSD-RM.

Several specialized frameworks are anticipated and encouraged. The NCOIC has already started to develop some of these, although additional ones will be required. A brief description of each specialized framework under development is provided in the following sections.

3.1 Net-centric Services Framework

The Net-Centric Services Framework (NCSF) provides designers of service-oriented systems and their users guidance for selecting patterns that optimize net-centricity and interoperability between services and systems. The NCSF builds upon the concepts and principles of service oriented architectures (SOAs) to ensure interoperability between SOAs in a network centric environment.

3.2 Communications Framework

Not all interoperability is a matter of protocols but protocols are central to communications interoperability. The NCOIC Communications framework will build upon the concepts captured in the NIF version 1 to provide guidelines for achieving communications interoperability in a network centric environment. The Communications Framework is based on the Internet Reference Model and includes a special category of patterns called patterns for communications (PFCs). PFCs provide guidelines for interoperability for lower-level communications functions.

3.3 Semantic Interoperability Framework

The Semantic Interoperability Framework (SIF) examines information objects, information exchange requirements (IERs) and information exchange models to develop guidelines for seamless sharing of information throughout and between enterprises.
3.4 Information Assurance and Security Framework

Information assurance and security are critical requirements for any information system. The Information Assurance and Security Framework (IASF) captures the concepts, principles and processes necessary to safely and securely share information in a network centric environment.
A Appendix A - NIF Model

This chapter describes the NIF model, which comprises two models:

- The Framework of Frameworks (FoF) model
- The repository model

A.1 FoF Model (Meta-architecture)

A.1.1 Model Classification

Artifacts are any work products such as requirements, definitions, patterns, documents, models, source code files, deployment descriptors, test cases or scripts, and so on. Artifacts are used during the whole life-cycle according to methodology requirements to capture essential information that help to define, develop, test, maintain or withdraw the product. The NIF Framework Repository will provide access to all artifacts developed in the context of the NIF. In general an artifact is captured inside a file.

The NIF identifies the following classes of artifacts:

- Overarching Architecture artifacts that support top-down overarching process
- Patterns focused artifacts that support middle-out and bottom-up process,
- General purpose artifacts that serve both OAA and Pattern-focused artifacts.

The NIF defines and uses the primary artifacts shown in Table 6:

<table>
<thead>
<tr>
<th>Artifact Category</th>
<th>Contents</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operational Descriptions</td>
<td>Pattern Focused Artifact</td>
<td>Captures network centric context (using SCOPE dimensions)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Captures network centric functional requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- textual requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- use cases</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- scenarios</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- interoperability problem</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- network centric capability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Captures network centric non-functional requirements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- network-ready key</td>
</tr>
<tr>
<td>Artifact Category</td>
<td>Contents</td>
<td>Purpose</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Network Centric Pattern</td>
<td>Provides network centric patterns</td>
<td>Specify model-based operational or technical guidance that packages standards to achieve effective net-centricity.</td>
</tr>
<tr>
<td>PFC</td>
<td>Provides Patterns for Communications (PFCs)</td>
<td>A specific type of network centric pattern that is focused on communications.</td>
</tr>
<tr>
<td>Use Cases</td>
<td>Captures network centric use cases</td>
<td>Assists in identifying network centric capabilities.</td>
</tr>
<tr>
<td>Scenario</td>
<td>Captures network centric scenarios</td>
<td>Assists in identifying network centric capabilities.</td>
</tr>
<tr>
<td>Global Attribute</td>
<td>Captures “globalness” characteristics</td>
<td>Refines global operational characteristics (e.g., non-functional SCOPE dimensions) at the design level.</td>
</tr>
<tr>
<td>Principles</td>
<td>Provides network centric architecture</td>
<td>Recommends a solution based on tenets, best practices, etc.</td>
</tr>
<tr>
<td>Architecture Description</td>
<td>Captures architecture description model</td>
<td>Describes the way in which systems should be connected in order to achieve effective net-centricity.</td>
</tr>
<tr>
<td></td>
<td>- system view</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- technical view</td>
<td></td>
</tr>
<tr>
<td>Architecture Document</td>
<td>Context and architecture requirements</td>
<td>Provides a template that enterprise or system architects may use for describing a complete network centric architecture.</td>
</tr>
<tr>
<td></td>
<td>Architecture specification</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- principles</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- patterns</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- criteria</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Architecture risks and opportunities</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Architectural forecast synthesis</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Architecting decisions log.</td>
<td></td>
</tr>
<tr>
<td>Service</td>
<td>Captures service and service-</td>
<td>Provides interface(s) that can be</td>
</tr>
</tbody>
</table>
Some of the NIF artifacts are detailed further in the following sections.

## A.1.2 Operational Descriptions (OD)

### A.1.2.1 OD Role

The role of the OD is to identify and capture network centric operational context and requirements in order to prepare for the specification of overarching system architecture.

Note that the ODs assume that some prior global contextual and operational analyses have been done and that

1. General-purpose capabilities have been captured using a format such as an Initial Capability Document (ICD), a CONOPS, an OCD, a NAF Capability View, a SCOPE study or any other appropriate format.
2. A security and safety analysis has been undertaken to identify threats and relevant security and safety capabilities.
3. An NCOIC SCOPE analysis also has been conducted, mostly to state the network centric context and capabilities requirements.

The SCOPE analysis characterizes network centric goals:

- By identifying relevant network centric capability-independent dimensions. Identification is driven by the SCOPE model, which identifies a set of general-purpose dimensions.
- By identifying network centric capabilities. Network centric capabilities are mined from general-purpose capabilities by abstracting those general-purpose capabilities from organizational and system structures.
NIF does not enforce any methodology to support contextual and operational analysis. NIF merely recommends that general-purpose capabilities be better identified:

1. Defining capabilities that effectively support mission goals (Effect-based Operations).
2. Using a business process modeling (BPM) approach, but other approaches could work as well.

### A.1.2.2 OD Description

The OD must be focused on capabilities and shielded from organizational issues or from technical resources that may support them.

An OD is described in several informative sections, as follow:

1. Architecture Principles and Artifacts
2. Context
3. Scenario
4. Information Model
   a. NCOV-2: Operational Connectivity balance
   b. NCOV-3: Operational Exchanges balance
   c. NCOV-7: Operational Information Model
5. Existing Assets
6. Use cases
7. NCO Capabilities balance
8. NCO Interoperability Gaps & Problems balance
9. NCO Standards candidates list
10. NCO Patterns

### A.1.2.3 OD development

As a starting point, one or several interoperability scenario(s) are to be developed in order to drive analysis and to capture operational capabilities, nodes and their relationships during operations. Note that not every possible scenario can be described. Specific scenarios should be captured to address the primary interoperability and network centric challenges. Subject matter experts should capture the balance of the information exchange needs later.

Having identified nodes and their relationships, Operational View Products will capture balances of connectivity, exchanges and information between operational nodes that are directed by scenario(s)
1. The capture process will start with raw Operational Connectivity information. Missing
   connectivity information not identified by scenarios will be added there.
2. This information will then be refined and abstracted to describe and detail Operational Exchanges.
   Refinement will detail information contents. Abstracting will consist into grouping point-to-point
   relationships into point-to-many relationships.
3. Then, the information that characterize the domain and those that are exchanged during scenario(s)
   runs will be grouped and structured together to form a consistent Operational Information Model
   able to answer multiple queries and to be extended to host (future) additional needs. This
   Operational Information Model will be used later to build an overarching (system) data model. An
   example of such model is the NATO/JC3IEDM.

At this point, it will be wise to iterate from the beginning in order to check for inconsistencies between
views and to check for incompleteness.

In order to further identify the specific NCO Capabilities that are required, a balance of NCO
interoperability gaps and problems shall be stated. This step implies that the user review actions already
performed to assess the value and feasibility of the proposed NCO solutions and to identify areas for
improvement.

A list of specific NCO capabilities can then be extracted from the general capabilities using the following
categories as guidelines:
   - Directing and controlling capabilities
   - Planning capabilities
   - Organizing capabilities
   - Adapting capabilities
   - Coordinating capabilities
   - Executing capabilities
   - Support services capabilities
   - Resource support capabilities
   - Informing capabilities
   - Communicating capabilities
   - Access and security capabilities
   - Information management capabilities

Capabilities can be described using NAF NCV-2 and NCV-4.
Finally, from that list of capabilities and after establishing a list of candidate operational standards, NCO patterns can be identified then extracted (mined) using similarities and commonalities between capabilities.

### A.1.2.4 GA: Global Attributes

Global attributes are simply the set of SCOPE dimensions that characterize the current context and so apply to the entire design.

### A.1.2.5 Scenario

For the network centric environment, ‘scenario’ is used at an operational level as a way to describe a set of events, actions or situations—for example, “search and rescue scenario” in a disaster response situation. Scenarios don’t define the mechanisms for collaboration or networking, but rather define an environment and situation in which the networking strategies to support overall goals are articulated.

A scenario is a synthetic description of an event or series of actions and events; it is also an account or synopsis of a projected course of action, events or situations. Scenario development is used in policy planning, organizational development and, generally, when organizations wish to test strategies against uncertain future developments. Scenarios are widely used by corporations to help understand different ways that future events could unfold. Scenario planning or scenario thinking is a complex business process related to futures studies.⁹

The informational focus is on describing an operational scenario in which collaborating entities require and use various network centric capabilities. Subtopics include:

1. Dynamic vs. static organization of resources for scenarios
2. Defined goals of scenarios
3. Postulated types of situational environment scenarios
4. Agreed upon roles and capabilities for collaborating entities
5. Special non-functional performance requirements

### A.1.2.6 Communications Scenarios

Operational scenarios model both information exchanged and the exchange process/modes between system elements.

- Information exchanged will help to define the data model later

---

Exchange modes will help to select most appropriate protocol later. Information exchanges can be captured with a modified OV-3 sub view. Exchange modes can be captured with a sequence diagram.

A.1.2.7 Asset Descriptions

System elements are the assets used by military, intelligence and civilian personnel to implement use cases that are needed to fulfill mission timelines. The term system element describes any equipment that can be linked to the network centric environment. This includes, but is not limited to the following: individually carried equipment, ground, vehicular and space-based platforms, sensor and communications networks, weapon systems, logistics, supplies, etc. Some system elements may be reused across many use cases. System elements contain one or more user applications that include communications components as part of their functions. A user application (UA) is defined as a collection of hardware and software that implements some function within a system element. Figure A-1 illustrates this model for a System Element.

![System Elements are composed of User Applications](image)

UAs contained in system elements and the PFCs defined in the technical view (TV) are distinct. A UA is a component of a system element that uses one or more communications-oriented PFCs as part of its functions. The PFCs are part of the NIF TV shown in each UA in figure A-1. The following list provides examples of the types of user applications targeted by the NIF:

- Intelligence gathering and analysis
- Situation awareness
- Command and control
- Positioning
- Targeting
- Weapon delivery
- Repair and maintenance management
- Deployment and logistics management
- Supply management

This list is not comprehensive; it can and will expand over time.
A.1.2.8 Use Case

For the network centric environment, “use case” has scope at an operations or business level that comprises a subset of the network operations that support a scenario. (Example: “allocate and activate search teams with the necessary resources to search for stranded people in a geographic area.”) At the network system level, one or more use cases are defined to focus on the informational, service, communication, and IA aspects supporting that business use case (Example: “search team information use case” identifies and specifies the information that the network should store, provide access to and enable the sharing of, in order to support the “allocate locate search team business use case.”)

- The business use case is described in technology-free terminology, which treats the business process as a black box and describes the business process by its business actors (people or systems) to achieve their goals (e.g., manual payment processing, expense report approval, management of corporate real estate.) The business use case will describe a process that provides value to the business actor and describes what the process does.
- The “network” system use cases are normally described at the sub process level (e.g., “create voucher”) and specify the data input and the expected data response. The system use case will describe how the actor and the system interact. For this reason it is recommended that a system use case specification begin with a verb (e.g., create voucher, select payments, exclude payment, cancel voucher.)

Subtopics for a use case include:
1. Identification of actors and their roles and capabilities
2. Top level flow of networking events, interactions, and activities comprising the collaborative use case
3. Top level start and end definitions for the use case

A use case provides a functional breakdown of a concept of operations (CONOPS) and establishes relationships between external actors and functions; specific scenarios can then illustrate them. Asset descriptions provide a framework for associating communication and data requirements with physical pieces of equipment--generally collections of military or civilian system elements such as: various types of combat vehicles, first-responder equipment, communications equipment, naval and space-based assets, etc. An operational scenario provides a sequenced breakdown of a CONOPS and lists the set of assets required to implement it.

A.1.2.8.1 Communications Use Cases

Defense and Intelligence applications typically begin their life with a CONOPS. CONOPS are broad-brush strokes that describe how various assets are assembled to accomplish missions. From these CONOPS, two relevant NIF components are derived: use cases and system elements.
Use cases represent a decomposition of a CONOPS mission timeline (or “mission thread”). A sequential and parallel aggregation of use cases is executed to implement a mission timeline.

Figure A-2 graphically represents the decomposition of a CONOPS into systems elements.

![Diagram of CONOPS and use cases](image)

**Figure A-2: The Relationships Between CONOPS, Use Cases, and Systems Elements**

System elements model the communication nodes in a network centric environment. Communication and data interoperability are implemented primarily by the software contained in the systems elements. A cornerstone of system elements’ collaboration in implementing use cases is insuring that communications protocols and data used by all system elements interoperate, as required.

### A.1.2.8.2 Use Case Format

Figure A-3 provides an example of use case description.
A network centric operation capability has scope at the operational level where interdependent networking relationships occur and, therefore, enable a capability that no single entity could achieve alone. To enable the desired effect, the network centric capability relies on patterns of communication, information sharing, knowledge acquisition, collaborative decision making, services offered by networked entities, and other networking aspects. The effect may be dynamically structured or statically pre-defined. Although network centric capabilities derive inductively from examinations of domain-specific scenarios and use cases, they can be generalized to support similar network centric capabilities, across multiple domains (for example, information sharing to support knowledge acquisition and decision making for a collaborative team performing a set of operations to achieve mission goals.)

Subtopics include:

1. The type of networking collaboration, e.g., joint decision, sharing information, achieving common information knowledge about the collaborative environment and resource status, requesting and committing to joint action
2. Optimizing allocation of collaborative resources
3. Coordinating resource allocation to reflect environmental changes
4. Identifying events that may affect successful operations
5. Event-triggered communication alerts and information sharing
6. Dynamic team formation
7. Notification of operation status—start, in-process, completed
8. Identification of resources necessary for operation

A.1.2.10 Network Centric Operation Interoperability Problem

This category focuses on interoperability problems due to operations at the organizational level which prevent or reduce the performance of a network centric capability. The impact of a network centric interoperability problem can be analyzed via a set of relevant NCOIC SCOPE model dimensions. A network centric interoperability problem at the operational level may result from an interoperability problem within the scope of one or more specialized frameworks, e.g., communications, services, information, knowledge management / semantic interoperability or information assurance.

Subtopics include:

1. Information sharing - not all participants can share specific operation-related information with collaborative participants, throughout the operation lifecycle
2. Communications - not all participants can communicate with other participants to coordinate operations activities
3. Service commitment - not all participants can commit to services in support of collaborative operations
4. Organizational - policies and safeguards inhibit collaborative operations across organizations

A.1.3 Principles

Principles are the overall recommendations and/or requirements that should be applied when using an asset or artifact to foster net-centricity. Just as using (open) standards is not enough to attain interoperability, using only network centric patterns is not enough to assure net-centricity, if those assets are not assembled using the overall rules that apply to the set of networked elements named in the overarching architecture.

A.1.4 Processes & Tasks

Processes and tasks are units of work that—under control—transform inputs into outputs, using resources:

- Inputs provides the program-specific information needed to achieve the process or task
• Outputs are information produced after processing
• Resources are tools and background information that support one or more processes or task
• Controls are information that drive the process or task

A process may or may not involve people, whereas a task is a unit of human work. A processes or task shall be described using a process/task name, inputs, outputs, controls and resources, as shown in Figure A- 4.

![Diagram showing process inputs, controls, resources, and outputs](image)

**Figure A- 4: Describing a Process/Task**

### A.1.5 Interfaces

The NIF considers the following categories of interfaces

• Physical interfaces (hardware interfaces)
• Logical interfaces
• Human system interfaces (physical + logical)

NIF recommends describing key interfaces using

• Static description,
• Dynamic description.

When available for the interface category, NIF recommends using an interface description language (IDL) to describe the interface.
At a minimum, a static description of an interface should include the information shown in Table A-1.

### Table A-1: Static Interface Description Elements

<table>
<thead>
<tr>
<th>Interface Element</th>
<th>Element Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customers</td>
<td>Interface’s actors</td>
</tr>
<tr>
<td>Function/Service ID</td>
<td>Function/service identifier</td>
</tr>
<tr>
<td>Operations</td>
<td>List of available operations</td>
</tr>
<tr>
<td>Operation Name</td>
<td>Name of operation provided by the function/service.</td>
</tr>
<tr>
<td>Operation Parameters</td>
<td>Description of operation’s parameters:</td>
</tr>
<tr>
<td></td>
<td>• Parameter name</td>
</tr>
<tr>
<td></td>
<td>• Parameter type</td>
</tr>
<tr>
<td></td>
<td>• Parameter value(s) or interval when applicable</td>
</tr>
<tr>
<td></td>
<td>• Parameter unit when applicable</td>
</tr>
<tr>
<td>Operation Results</td>
<td>Description of operation’s results:</td>
</tr>
<tr>
<td></td>
<td>• Result name</td>
</tr>
<tr>
<td></td>
<td>• Result type</td>
</tr>
<tr>
<td></td>
<td>• Result value(s) when applicable</td>
</tr>
<tr>
<td></td>
<td>• Result unit when applicable</td>
</tr>
<tr>
<td>Exceptions</td>
<td>Description of possible expected failures</td>
</tr>
</tbody>
</table>

An interface’s dynamic description should illustrate at least one of the scenario types shown in Table A-2.

### Table A-2: Dynamic Interface Description Types

<table>
<thead>
<tr>
<th>Scenario type</th>
<th>Scenario Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal operation</td>
<td>Sequence(s) of interface’s operations calls and responses during normal operation(s) processing</td>
</tr>
<tr>
<td>Fault operation</td>
<td>Sequence(s) of interface’s operations calls and responses when a fault occurs</td>
</tr>
</tbody>
</table>
### Scenario Type and Description

<table>
<thead>
<tr>
<th>Scenario type</th>
<th>Scenario Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Init scenario</td>
<td>Sequence(s) of interface’s operations calls and responses during initialization</td>
</tr>
<tr>
<td>Management scenario</td>
<td>Sequence(s) of interface’s operations calls and responses during management</td>
</tr>
</tbody>
</table>

Scenario descriptions should detail pre-and-post conditions.

#### A.1.6 Network Centric Views

A view represents the whole system from a particular stakeholder or user’s point of view. A view reduces perceived complexity through separation of concerns.

Architecture views are an end product of the process of “architecting.”

When feasible, NIF recommends addressing and using the following views from a network centric point of view:

- Context view (e.g., ontology, glossary, program risks, forecasts)
- Capability view (e.g., capability phasing)
- Program view (e.g., procurement synchronization)
- Operational view (e.g., ODs)
- Service View (e.g., services, SLA, QoS, interfaces)
- System view (e.g., system-level services)
- Technical view (e.g., PFCs)
- Deployment view (e.g., Blueprints)
- Information view (e.g., Information model)
- Security view (e.g., threats, vulnerabilities, security functions …)
- System management view (e.g., managed and management objects)
- Safety view

All views should be consistently defined and linked.

Views should be extensible and based upon standard architecture frameworks when relevant.

### NIF and Architecture Frameworks

Most architecture frameworks (AFs) such as DoDAF, MoDAF and NAF provide a way to describe architecture views and products. Other AFs such as TOGAF and AGATE provide process and methodology guidance.
NIF does not prescribe any specific architecture framework; it does recommend specifying architectures using relevant elements of AFs to build network centric architectures. As an example, the NIF recommends developing in whole, or in part, the set of end-products (views) described in Table A-3.

<table>
<thead>
<tr>
<th>View Product</th>
<th>Name</th>
<th>Recommendation Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCAV</td>
<td>Network centric Architectural Views (NCAV)</td>
<td></td>
</tr>
<tr>
<td>NCAV-1</td>
<td>Overview and Summary Information</td>
<td></td>
</tr>
<tr>
<td>NCAV-2</td>
<td>Integrated Dictionary</td>
<td>A formal model is highly recommended to define: 1. Development artifacts 2. Domain information model</td>
</tr>
<tr>
<td>NCOV</td>
<td>Network centric Operational View (NCOV)</td>
<td></td>
</tr>
<tr>
<td>NCOV-2</td>
<td>Operational Connectivity Description</td>
<td>From a network centric point of view, this product must be considered as a provision.</td>
</tr>
<tr>
<td>NCOV-3</td>
<td>Operational Information Exchange Matrix</td>
<td>IERs must be considered as necessary but not sufficient requirements: a. to define interfaces between nodes, b. to build the overall logical data model.</td>
</tr>
<tr>
<td>NCOV-7</td>
<td>Logical Information Model</td>
<td>Logical Information Model</td>
</tr>
<tr>
<td>NCISOV</td>
<td>Network Centric Service Oriented View (NCISOV)</td>
<td>To be used when service-oriented approach is selected in whole, or in part.</td>
</tr>
<tr>
<td>NCISOV-1</td>
<td>Service Definitions</td>
<td></td>
</tr>
<tr>
<td>NCISOV-2</td>
<td>Service Orchestration</td>
<td></td>
</tr>
<tr>
<td>NCSV</td>
<td>Network Centric System View (NCSV)</td>
<td></td>
</tr>
<tr>
<td>NCSV-1</td>
<td>Systems Interface Description</td>
<td></td>
</tr>
<tr>
<td>NCSV-2</td>
<td>Systems Communication Description</td>
<td></td>
</tr>
</tbody>
</table>
A.1.6.1  **NCOV : Network Centric Operational View**

NIF recommends that the NCOV operational view be used as part of the operational description (OD)--specified earlier in this document.

A.1.6.1.1  **NCOV-2 : Network Centric Operational Connectivity**

**Purpose**

The purpose of the operational node connectivity description is to illustrate the operational domain’s needs for information exchanges in support of operational activities.

NCOV-2 depicts operational nodes with “need lines” that connect those nodes that indicate a need to exchange information. An NCOV-2 may be annotated to show the flows of materiel, energy or people between nodes (note that these exchanges are not need lines and do not appear in an NCOV-3). In order to promote a network centric operational approach that is independent from organizational structures and resources types, NIF recommend building the NCOV-2 in 2 steps:

-  Step one: capture “raw” required information extracted from scenario. This information is scenario-specific
-  Step two: abstract required information to focus on operational activities and to shield organization structures and resources
A.1.6.1.2    NCOV-3 : Network Centric Information Exchanges

**Purpose:** the goal of the NCOV-3 is to gather the details of information exchange requirements (IERs):
- IER description and type.
- Requesting node
- Supplying node
- Exchange description and constraints

An IER describes the contents of Information Exchange.
- Requesting nodes determine the node types that may request the IER
- Supplying nodes determine the nodes that should provide the IER
- Exchange description includes:
  - Information description
  - Information context
- Exchange constraints are those non-functional constraints that apply to IERs such as:
  - Performance
  - Security
  - Safety
  - Others (use relevant SCOPE dimensions as a source)

NCOIC emphasizes that IERs must not be limited to actual point-to-point needs based on a limited set of scenarios, but that they be extended and abstracted, in order to support one-to-many exchanges.

NIF requires no specific format but suggests that users reuse the NCOV-3 format which is extended to account for information context and SCOPE attributes. These are shown in Figure A-5.
### OV-3 Operational Information Exchange Matrix for Emergency Medical Service

<table>
<thead>
<tr>
<th>Operational Information Elements</th>
<th>Information Source</th>
<th>Information Destination</th>
<th>Information Exchange Attributes from SCOPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Description</td>
<td>Media</td>
<td>Size/Units</td>
</tr>
<tr>
<td>Automatic Accident Notification</td>
<td>provided by CR team</td>
<td>Text</td>
<td>provided by CR team</td>
</tr>
<tr>
<td>Accident notification</td>
<td>provided by CR team</td>
<td>Text</td>
<td>provided by CR team</td>
</tr>
<tr>
<td>Safe route</td>
<td>provided by CR team</td>
<td>Map</td>
<td>provided by CR team</td>
</tr>
<tr>
<td>Medevac mission</td>
<td>provided by CR team</td>
<td>Text</td>
<td>provided by CR team</td>
</tr>
<tr>
<td>Accident location</td>
<td>provided by CR team</td>
<td>Text</td>
<td>provided by CR team</td>
</tr>
<tr>
<td>Medical evaluation</td>
<td>provided by CR team</td>
<td>Text</td>
<td>provided by CR team</td>
</tr>
<tr>
<td>Accident scene</td>
<td>provided by CR team</td>
<td>Video</td>
<td>provided by CR team</td>
</tr>
<tr>
<td>Electrocardiogram</td>
<td>provided by CR team</td>
<td>Data stream</td>
<td>provided by CR team</td>
</tr>
<tr>
<td>Skull scan</td>
<td>provided by CR team</td>
<td>Image</td>
<td>provided by CR team</td>
</tr>
</tbody>
</table>

**Figure A-5: Example NOV-3 IER Matrix**
A.1.6.1.3 NCOV-7: Network Centric Information Model

Definition:
Differing from a data model, an information model represents domain knowledge, according to its information aspect. In other words, it is a model of information about concepts in the universe of discourse, relevant to the architecture effort. For example, if the operational domain recognizes the existence of a concept called “weapon platforms,” then the information model would contain information objects that reflect what we want to know about weapon platforms and what we want to communicate about weapons platforms to others. As such, the information model is inherently conceptual in nature.

Purpose:
The NCOV-7 goal is to capture an overall (i.e., System-of-Systems) information model that can be used by the system view to consistently define system interfaces (i.e., using the same data for the same information).

The NCOV-7 must be initiated with information contained in IERs, but from a network centric point-of-view, the IERs should not be limited to those that have been identified through scenarios and other artifacts (NCOV-3). Future IERs must be anticipated and provisions made for future needs. Additionally, at the information exchange level, operational exchanges must not be considered as point-to-point exchanges but as point-to-many exchanges that operates on a global logical information model. In other word this means that the network centric information model is not just the sum of information identified by Information Exchange Requirements (IERs) but must be designed as a global consistent information model supported by a global consistent metamodel.

No specific format is required by NIF but NIF suggest reusing NOV-7 format.

A.1.6.2 NCSOV: Network Centric Services-Oriented View

NIF recommend using service-oriented views when a service-oriented approach is selected. The services concept will be detailed in the Network Centric Services Specialized Framework, currently under development by NCOIC.

A.1.6.3 NCSV: Network Centric System View

NIF recommends using system views to describe systems participating in a network centric grouping. The system view will be part of the overarching architecture specification.
A.1.6.4   NCTV: Network Centric Technical View

NIF recommends using the technical views for describing overall technical elements. The technical view will be part of the overarching architecture specification.

A.1.7   Network Centric Patterns

The Network Centric Pattern (NCP) concept was introduced in Section 1 of the NIF™ Solution Description Reference Manual (NSD-RM). This appendix specifies the content and format for creating NCPs and helps to ensure uniformity and consistency. An NCP captures best practices for creating systems with network centric capabilities. These capabilities mitigate specific network centric interoperability problems. NCPs are templates that provide solutions for network centricity and verifiability.

NCOIC may receive patterns from different sources, both from within and outside of NCOIC. For example, working groups within the NCOIC Specialized Frameworks Functional Team will generate technical NCPs and the NCOIC integrated project teams, which focus on customer missions will develop operational or capability NCPs for a mission domain. All NCPs must abide by network centric principles. Technical NCPs must have measurable and testable criteria by which products & services can be verified conformant with the pattern. Information on how network centric products can be certified conformant is the responsibility of the NCOIC Building Blocks team.

Sections in this Appendix include an Overview, NCOIC NCP categories, and a Common Description.

A.1.7.1   Introduction

The NCP methodology is similar to that commonly used for example in Design Patterns\(^{11}\), with the addition of NCP attributes that focus on network centricity. A general definition of a pattern is:

\"Anything proposed for imitation; an archetype; an exemplar; that which is to be, or is worthy to be, copied or imitated; as, a pattern of a machine.\"\(^{12}\)

The pattern definition as used in NCOIC is described in Section 1 of the NSD-RM main document.

Behavioral patterns play a central role in human activity. We all discover certain methods or procedures that work well in terms of solving certain life problems. We learn many patterns from others, and some we create by trial and error. Successful patterns are often reused, and adapted, to suit new problems or situations. Patterns should be based on successful net-centric applications rather than developed without any previous successful applications. In this way, patterns may be used to develop a proposed solution to a problem with greater confidence in, and understanding of, its applicability to the problem.

The discipline of patterns was first developed for architecture; as guides for designing buildings, rooms, towns and other physical systems, and as such, proved to be a useful aid.\(^{13}\) Patterns also proved

\(^{10}\) See section A.1.7.3 for a full definition of the Technical, Capability, and Operational NCP categories

\(^{11}\) Gamma, Helm, et.al, Design Patterns, Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995

\(^{12}\) Webster's Revised Unabridged Dictionary (1913)
valuable in developing software systems. There are many canonical software architecture patterns; client-server, as an example. Leveraging the pattern concept, NCOIC uses NCPs to describe well-known network centric solutions to problems as a means of providing guidance to architects and developers of network centric systems. In the context of NCOIC’s specialized frameworks, NCPs will focus on critical operational, capability, and technology areas such as communication and information assurance. NCOIC is dedicated to providing NCPs for developers that meet the following objective:

“A Net-Centric Pattern (NCP) provides expert guidance based on standards and net-centric principles for creating systems to mitigate specific Net-Centric interoperability problems.”

A.1.7.2 NCOIC Network Centric Patterns (NCP)

As used by NCOIC, a pattern is a network centric solution to a problem in a given context. An NCP is a document that provides expert guidance based on experience and standards that can be used for creating systems with desired network centric capabilities and mitigating specific network centric interoperability problems. An NCP follows and adheres to the following network centric characteristics:

- An NCP is a pattern that solves a clearly stated problem. An NCP covers one or more network centric capabilities and interoperability problem solutions. It describes how a network centric capability is enabled and problem mitigated. It provides guidance in the use of best practices in the form of principles, tenets and standards to enable these capabilities and mitigate interoperability problems.
- Each NCP conforms to a set of common pattern attributes, plus any unique Specialized Framework attributes that refine or complement NIF Global Attributes for a particular technical domain.
- NCP attributes may have a formal, semi-formal or informal description. When possible, NCPs should be described with a formal or semi-formal description using modeling language standards such as the Object Management Group (OMG) Business Process Modeling Notation (BPMN), OMG Unified Modeling Language (UML), and OMG Systems Modeling Language (SysML), Integration Definition (IDEF), or other descriptions or languages appropriate to the NCP’s domain. The various views as represented in Architectural Frameworks (e.g. NAF, DoDAF, TOGAF, MoDAF, AusDAF) can also be useful in capturing the content for a given NCP attribute.
- Technical NCPs must include attributes that are measurable and can be tested. Operational and Capability NCPs should utilize SCOPE model dimensions and values to provide semi-quantitative attribute value ranges to describe the operational or capability “space” for which the NCP is intended to be used and over which the NCP is considered to be applicable.

---

13 Christopher Alexander is known as the innovator of this approach. See www.patternlanguage.com for more information.
14 Hans Polzer, presentation to NCOIC general Plenary session, June 2, 2009.
15 Context can be an operational description, scenario, use cases, SCOPE dimensions, and environmental factors.
16 NCOIC’s Net-Centric Services Framework chapter 4 describes Net Centric Patterns in the context of net centric services.
17 Various Architectural Frameworks from NATO (NAF), the U.S. Department of Defense (DoDAF), The Open Group (TOGAF), the U.K. Ministry of Defense (MoDAF), and The Australian Department of Defense (AusDAF).
18 See section A.1.7.3 for a full definition of the Technical, Capability, and Operational NCP categories.
NCPs support service-oriented architectures. The **structure** attribute described below can be used to describe the services provided within the NCP, and the **behavior** attribute below can be used to describe how these services can be used to solve the targeted problem.

NCPs must support and comply with the net-centric principles contained in the NIF NSD-RM (this document) as well as the principles contained in applicable Specialized Frameworks. Section A.1.7.5.3 provides additional information on capturing the mapping between NCPs and principles.

### A.1.7.3 NCP Categories

NCPs are architecture “fragments” and some of these fragments contain more emphasis on certain architecture view types than others, as well as some coupling between view types. The NCPs represent a “partial disintegration” of architecture into more generic components that can be re-assembled and tailored to support a variety of integrated architectures.

The three NCP categories defined by the NIF-RM are Operational, Capability, and Technical:

- **Operational NCP**: Describes typical interoperability issues in an operational mission context and provides guidance to mitigate these problems. An Operational NCP captures standard operational practices, identifies relevant operational capabilities and their interoperability requirements. These are needed to conduct activities (military operations or business objectives) in a given mission context. An Operational NCP uses operational concepts and terminology. It is the final outcome of an operational analysis whose results can be captured using an Operational Description. An example of an Operational NCP could be a force deployment planning pattern.

- **Capability NCP**: Refines operational capabilities or NCPs. It describes operational architecture and scope issues and associated system functionalities that mitigate systems interoperability problems when implementing a capability in a net-centric environment. A Capability NCP describes actor types, use cases, process flows and functions that support interaction among the actors to achieve specific results/effects. A Capability NCP may be motivated by, and used in, one or more operational NCPs. The Operational NCPs provide context and scope for the Capability NCP. The Capability NCP is a functional architecture fragment that shows how one or more Operational NCPs may be supported by the Capability NCP, often with the support of other Capability NCPs. Capability NCPs provide context/scope framework for design of Technical NCPs that use appropriate network-centric standards for implementation of the Capability NCP. A capability also allocates functional capabilities to hardware, software and human actions. It may also identify relevant domain standards, as well as Key Performance Parameters such as Quality of Service and availability. A Capability NCP example might be an airlift scheduling pattern—used to support a force deployment planning Operational NCP, a force sustainment Operational NCP or a disaster relief Operational NCP.

- **Technical NCP**: Describes technical standards, technologies, and interoperability techniques that support capabilities in the functional context specified in a Capability NCP. A Technical NCP

---

19 For example, deriving an operating picture whose elements are operational-level assets or artifacts such as Universal Joint Task Lists (UJTLs), unit types, DOTMLPF (Doctrine, Organization, Training, Materiel, Leadership, Personnel, Facilities) elements, or civilian equivalent.
describes design elements based on the capabilities that support operational mission(s), e.g. how to safely access a core network whose elements are system-level or technical-level artifacts, such as services or components. Capability NCPs may leverage Technical NCPs to describe a solution.

There may be cases where Technical NCPs make some implicit capability scope assumptions, such as use of a particular organization’s infrastructure services and standards (and associated operational scope limitations), but this should be avoided if possible, and explicitly stated in the technical NCP pre-conditions as a NCP scope assumption. An example might be a Technical NCP intended to operate on a specific military network, but otherwise not specific to any particular capability scope.

Figure A-6 also shows relational dependencies between Operational, Capability, and Technical NCPs. Operational NCPs are the “big picture” patterns, where Capability NCPs provide support for an operational NCP. Each capability NCP may be further decomposed and supported by technical solutions presented in technical NCPs as noted above.

A.1.7.4 Common Description

An NCP is intended to be an implementable and reusable artifact, providing unambiguous direction. It is written in a prescriptive language as described in Section A.1.7.5 so that the instantiated service or application may be verified by NCOIC. It uses standardized graphics (i.e. Business Process Modeling Notation (BPMN), Universal Modeling Language (UML), or System Modeling Language (SysML) for drawings, but also may contain other figures and tables for clarity. This generic definition is extensible to other specific frameworks. For example, an NCP defined within the Mobility Framework includes elements listed below but may include other elements specific to the Mobility domain. NCP document composition should be governed by the following rules:
• Limit to a reasonable number of pages (e.g. less than 30)
• Avoid compound NCPs (multiple distinguishable topics in a single NCP)
• Avoid overlap with existing NCPs by providing references to those NCPs

The following standard outline describes and defines the contents of an NCP. This standard outline should be used for all NCOIC NCPs: Table A-4-1 describes the outline for the cover page

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Format</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>This cover page provides for organization and retrieval, thereby increasing use. This cover page outline assists retrieval of a pattern from a Pattern Database, such as a Reusable Asset Specification or other potential data model. The cover page could extend to a second page if version history or style dictates.</td>
<td></td>
<td>Cover Page is Required</td>
</tr>
<tr>
<td>Title (Pattern Name)</td>
<td>Provide a descriptive &amp; unique name that helps in identifying and referring to the NCP. Include the NCP category in the name (“Operational,” “Capability,” or “Technical”), e.g., “XYZ Technical NCP”</td>
<td>Free text</td>
<td>Required</td>
</tr>
<tr>
<td>Aliases (Part of Title)</td>
<td>List alternative names, if any, that may be (or have been) used in place of the name listed above, as different nationalities or technical disciplines may know it by a different name.</td>
<td>Free text separated from primary title by parentheses</td>
<td>Optional</td>
</tr>
<tr>
<td>Author(s)</td>
<td>List the Author(s) and affiliation(s)</td>
<td>Free Text</td>
<td>Required</td>
</tr>
</tbody>
</table>
| Date & Version History        | • Version identifier for the NCP for configuration management containing a list of present and prior version numbers and dates. Do not include draft versions  
• Include a short description of changes, and compatibility with prior versions (fully, partially, or not compatible)  
• Defer detailed change descriptions to an Appendix  
• Sample version history might be: V1.1-2009-01-20 Standard xyz updated. Partially compatible with previous versions.                                                                                                                                                                                                                                           | NCOIC standard versioning, e.g. V1.1-2009-01-20 | Required
<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Format</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.0-2008-12-25</td>
<td>Clarified verification methodology. Fully compatible with previous versions.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abstract</td>
<td>Each NCP contains an abstract that summarizes the problem, course of action, results obtained, conclusions reached, and recommendations made in the briefest manner consistent with clarity (not more than 200 words) and designed for the convenience of library search and for homepage linkage. The abstract is informative, presenting the principal ideas, experimental techniques, and significant quantitative and qualitative data, rather than a general indicative statement that the report contains certain types or classes of information.</td>
<td>Text</td>
<td>Required</td>
</tr>
<tr>
<td>Header and Footer</td>
<td>Header: contains NCOIC logo, NCP name, name of NCOIC team or working group developing the document. Footer: contains page #, version #, &amp; clearance status, e.g. approved for public release, not approved for public release, working group internal, technical committee internal.</td>
<td>Use this document as an example</td>
<td>Required</td>
</tr>
</tbody>
</table>

A table of contents may be added after the cover page, followed by the NCP sections. Table A-4-2 describes the outline for Section 1 of the NCP. Section 1 is the context and problem description, and Section 2 contains the prescriptive solution including Actors, Interfaces, Pre-conditions, Structure, Behavior, Post-conditions, and Standards.

**Table A-4-2: Description Outline**

<table>
<thead>
<tr>
<th>Section/Sub-Section</th>
<th>Contents Format</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction and Problem Description</td>
<td>The following sub-sections define the NCP, its context, and the manner in which it is used. It provides the detail necessary for a user to apply the NCP.</td>
<td>Heading</td>
</tr>
</tbody>
</table>

Note that the pattern is a document that provides both direction and guidance to a developer. That developer would use this pattern for development of a physical service supplied to a user of the service.
<table>
<thead>
<tr>
<th>Section/Sub-Section</th>
<th>Contents Format</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.1 Context</strong></td>
<td>Describe the context or scenario in which the NCP is applicable, and conditions under which the NCP is to be used. Context includes the environmental entities with which the NCP may interact. Also state the environmental conditions required for this NCP to be applied. For an Operational NCP, describe the mission domain and the operational entities. For a Capability NCP, describe a scenario or use case that may explain both operational and technical entities. For a Technical NCP explain the technical goal and include the technical entities. Any net-centric principles from the NIF NSD-RM and applicable Specialized Frameworks supported by this NCP should be identified in this section. Keep this section short, with details in the precondition section or in an Appendix.</td>
<td>Free Text + figures (not prescriptive)</td>
</tr>
<tr>
<td><strong>1.2 Problem Statement</strong></td>
<td>Describe the problem the NCP solves within the above context, in other words, <em>why is this pattern needed?</em> Include the primary NCP goal and the reason for using it. Note that several domains can be addressed by a single NCP.</td>
<td>Free Text (not prescriptive)</td>
</tr>
<tr>
<td><strong>1.3 Expected Benefits</strong></td>
<td>Describe the advantages accruing from the use of this NCP. This paragraph should provide the users of the NCP a reason for following this pattern as opposed to other guidance or no guidance at all.</td>
<td>Free Text</td>
</tr>
<tr>
<td><strong>2.0 Recommended Solution</strong></td>
<td></td>
<td>Heading</td>
</tr>
<tr>
<td>Section/Sub-Section</td>
<td>Contents Format</td>
<td>Comment</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------</td>
<td>---------</td>
</tr>
<tr>
<td>2.1 Actors</td>
<td>Describe precisely the actors that require this operation/capability/technology to achieve their goal. Include indirect actors that may provide input, or receive output, as a result of NCP use, with an appropriate method to show the actors that are associated with the NCP. Views &amp; Viewpoints (pictures) are “worth a thousand words” if the viewer understands the visual methodology: o For an Operational NCP, describe how people and systems (both internal and external) interact to accomplish their activities (military operations or business objectives) in the given mission context o For a Capability NCP, describe how people and external systems invoke (or use or supply data to, or receive data from) the capability to achieve the required functions o For a Technical NCP, describe how people and external systems interact with the system or service that incorporates the NCP.</td>
<td>Example formats are found in section A.1.7.5.5 Required Sub-Section</td>
</tr>
<tr>
<td>2.2 Interfaces</td>
<td>Identify and describe specific characteristics of interfaces between various actors. Also identify interfaces between actors and the system(s) or services associated with the NCP. This section addresses static interfaces rather than dynamic interfaces, defined in Section 1.5, Behavior.</td>
<td>Tabular. Prefer an interface definition language Required Sub-Section</td>
</tr>
<tr>
<td>2.3 Pre-Conditions</td>
<td>Describe the prerequisites that must be in place before the NCP can be applied, as follows: o Operational NCP: Include a description of operational state(s) that must exist before the NCP can be applied o Capability NCP: Include descriptions of conditions that must be in place before the capability can be used o Technical NCP: Conditions that must be satisfied or in place for the prescribed solution to function properly If the stated pre-conditions are not met, the pattern cannot be successfully applied to the problem at hand.</td>
<td>Text Required Sub-Section</td>
</tr>
<tr>
<td>Section/Sub-Section</td>
<td>Contents Format</td>
<td>Comment</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>2.4 Structure</strong></td>
<td>The structure describes logical static elements supporting the understanding and the implementation of the NCP.</td>
<td>Example formats are found in section A.1.7.5.5</td>
</tr>
<tr>
<td></td>
<td>• Identify the scope of the NCP</td>
<td>Required Sub-Section</td>
</tr>
<tr>
<td></td>
<td>• Specify main objects (components, functions and services) used in this NCP and their roles and relationships in the design</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Distinguish between provided and consumed functions and services</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This attribute provides a description of the static structure present within the NCP.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Use an appropriate method to show the structure of the operation/capability/technology as applicable to its use, such as a block diagram of components</td>
<td></td>
</tr>
<tr>
<td><strong>2.5 Behavior</strong></td>
<td>• Describe the dynamic interaction of the structure elements, interfaces, and actors described in the prior sections. Describe major interactions required to achieve the intent of the NCP</td>
<td>Example formats are found in section A.1.7.5.5</td>
</tr>
<tr>
<td></td>
<td>• Describe the major steps to accomplish the operation or capability NCP or the sequence of activities and interactions of components in the Technical NCP</td>
<td>Required Sub-Section</td>
</tr>
<tr>
<td></td>
<td>• Suggest using an appropriate method to show interactions in the operation/capability/technology, such as flow diagrams of procedural steps.</td>
<td></td>
</tr>
<tr>
<td>Section/Sub-Section</td>
<td>Contents Format</td>
<td>Comment</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| **2.6 Post-Conditions** | Post-Conditions are the concrete results of applying the pattern. Describe the outcome, consequences, limitations, and any potential side-effects resulting from using the NCP  
- Operational NCP: Include descriptions of operational conditions resulting from the operational scenario  
- Capability NCP: Include a description of the conditions that result from the use of the capability  
- Technical NCP: include  
  - Limitations and/or consequences resulting from use of the NCP  
  - States and modes resulting from use of the technology  
Post-Conditions differ from Expected Benefits in that Expected Benefits describe the advantages of using the pattern, whereas Post-Conditions are the concrete results, both good and bad. | Text | Required Sub-Section |
| **2.7 Standards** | Describe required standard(s), including version(s) and optional variations applied in the operations/capability/technology implementation. Specified protocols should be associated with selected standard(s)  
- Use Open or Commercial Standards such as IETF, ITU, IEEE, ISO, OMG, AIAA  
- Do not use Proprietary Standards  
- Use DoDAF / MoDAF / NAF / AusDAF diagrams if appropriate  
- Minimize use of special or country unique defense standards e.g., US DoD or NATO classified or restricted distribution standards | Text & Tables | Sub-section if technical standards are required. |
Table A-4-3 contains the Implementation Guidance in Section 3. This section contains non-prescriptive Expert Advice, Lessons Learned, Constraints and Opportunities, NIF Guidance, Specialized Framework Guidance, Known Uses, Potential Capability, and References. Information that is required for NCP conformance with NIF tenets and principles is contained in Table A-4-3 under paragraph 3.3.

<table>
<thead>
<tr>
<th>Section/Sub-Section</th>
<th>Contents</th>
<th>Format</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. Additional</td>
<td>This section should contain addition, non-prescriptive information that</td>
<td>Heading</td>
<td></td>
</tr>
<tr>
<td>Information</td>
<td>can aid the system developers in applying the pattern.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1 Lessons</td>
<td>Describe expert advice, lessons learned, and best practices that</td>
<td>Text</td>
<td>As applicable</td>
</tr>
<tr>
<td>Learned</td>
<td>Subject Matter Experts (SMEs) recommend for successful implementation</td>
<td>(not prescriptive)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>of the standard(s) applicable to this NCP, especially:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>● Known cost / schedule / performance risks</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>● Interoperability failure Root Causes using specified standards in</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>prior and current systems (at time of release)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A-4-3: Implementation Guidance
<table>
<thead>
<tr>
<th>Section/Sub-Section</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2 Constraints &amp; Opportunities</td>
<td><strong>Constraints:</strong> This section captures any constraints on the solution recommended by the SMEs that may affect interoperability. Without constraints, a single set of standards would be used and implemented in the same manner. Reality dictates there are factors that constrain the use of simple solutions or limits the ability of certain solutions to achieve interoperability. The recommendations by the NCP author (i.e., the SMEs) might include invoking certain optional behavior or features of the standards; implementing more than one standard; providing gateways, bridges, or translators; or possibly even noting that certain types of users/systems might be limited in their ability to interoperate if a reasonable solution was not available. Other possible examples of &quot;constraints&quot; might include: performance, security and safety requirements/considerations. <strong>Opportunities:</strong> Opportunities capture ways to achieve significant improvements in interoperability over previous/other approaches, or to identify ways to mitigate some of the restrictions imposed by the constraints. For example, wrapping an existing legacy system to provide services in a SOA enterprise without having to redesign the legacy system.</td>
</tr>
<tr>
<td>3.3 Known Uses</td>
<td>List missions and/or systems which use or have used the approach recommended in this NCP</td>
</tr>
<tr>
<td>3.4 Potential Capability</td>
<td>Identify any portions of the NCP which can have increased capability without interfering with interoperability: - Operational NCP: may include mission flexibility - Capability NCP: may include capability variation - Technical NCP: may include additional capability (such as use of an optional extension of a standard) which if excluded or included would not impact interoperability of systems incorporating Building Blocks from different vendors. Also refer to section A.1.7.5.4</td>
</tr>
</tbody>
</table>

**Format** | **Comment**
---|---
Text | Required
Text or Table | Optional. Also refer to section A.1.7.5.4.
Text | Optional. Also refer to section A.1.7.5.5.
<table>
<thead>
<tr>
<th>Section/Sub-Section</th>
<th>Contents</th>
<th>Format</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5 Related Patterns</td>
<td>Identify any related NCPs or associated NCPs</td>
<td>Text</td>
<td>Related NCPs required, Associated NCPs optional</td>
</tr>
</tbody>
</table>
| 3.6 References | • Identify any industry-standard documentation, reports, or other materials that designers may find useful in designing missions or systems that incorporate the NCP  
• Keep this section short by just providing links  
• Place details in Appendices | Text or Table | Optional |
Table A-4-4 describes the outline for Section 3 (Verification). This paragraph verifies that building blocks based implementations are conformant with the NCP. Section 3 of the NCP contains information that enables any service or capability (i.e., Building Blocks) to be developed such that it can be verified as compliant with the NCP in accordance with the NCOIC Building Blocks certification policies and procedures.

Table A-4-4: Verification

<table>
<thead>
<tr>
<th>Section/Sub-Section</th>
<th>Description Format</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. Verification</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This section’s intent is to provide Building Blocks suppliers a minimum set of requirements to ensure that their solutions conform to the guidance, standards, specifications and protocols contained within the NCP. It also identifies which methodologies and indicators of conformance the Building Blocks suppliers must provide to show conformance with the pattern for NCOIC Building Blocks certification. Verification must be accomplished via specified methods such as Analysis, Test, Demonstration, Inspection, etc.

This section should indicate the allowable methods as well as a definition for each method listed.

Tabular

To facilitate verification, the style of the supplier’s product description must be similar to that used in commercial or military requirements documents. Table identifiers must contain references to the prescriptive sections of the NCP, and must contain normative references to the sections of the standards, specifications, and protocols that are applicable.

Refer to Section A. 1.7.5.5 for additional recommendations on the content and format for the table.

Required for Technical NCPs.
Optional for Capability NCP (by demonstration or inspection).
Not applicable for Operational NCPs.

It is important to verify requirements only for products or services that implement this NCP, not the entire system.

This section is the link between NCPs and Building Blocks.

In Section 4, requirements distinguish those elements of an NCP or a Building Block - an instantiation of the guidance contained in an NCP - that can be verified. These requirements are an essential part of the NCOIC Building Blocks Certification process. Per the CMMI21, “Validation ensures that the product actually meets the user's needs, and that the specifications were correct in the first place, while verification is ensuring that the product has been built according to the requirements and design

---

specifications.” In the NCOIC context, the following definition applies: Patterns are validated to ensure they meet operational needs and requirements and contain the correct standards for proper functioning as stated in the pattern. Building Blocks (products produced using patterns) are verified for conformance with the requirements contained in the patterns.

These guidelines should be followed when describing requirements in this section:

- Define requirements using “shall”, “must”, “should”, or “may” statements. “Shall” and “must” indicate mandatory conformance to the requirement. “Should” indicates a recommended approach, but is not required for conformance purposes. “May” indicates allowable behavior, but is not considered for conformance purposes.

- Use the following examples as the basis of requirements:
  - Actors and Use Cases on Use Case diagrams
  - Interfaces in SysML / UML / IDEF diagrams
  - DoDAF / MoDAF / NAF / AusDAF models
  - SCOPE model dimension values resulting from a SCOPE analysis of the OD or NCP
  - NIF Overarching & Specialized Framework Principles (e.g., service litmus test, section 5 of Network Centric Services Framework)

- Use metrics in requirement statements where applicable

- Include in Capability NCP metrics that describe Business Objectives and/or Measures of Effectiveness (MOE) and/or Measures of Performance (MOP)

- Technical NCP metrics should include MOPs and/or Technical Performance Measurements (TPMs)
  - Metrics are deterministic and not subjective
  - Metrics are verifiable via analysis, test, demonstration, and/or inspection

- Operational NCPs should include Critical Operational Issues (COIs)

- Identify interdependencies between metrics (e.g., For technical NCPs, Output Power level vs. Bit Rate)

- Include requirements for optional extensions in this section. For example, a Technical NCP regarding communications may include an objective bit rate (higher than a threshold or minimum bit rate); Therefore, this NCP would include:
  - One requirement statement for the threshold or minimum required bit rate
  - A second requirement statement for the objective or higher bit rate

- Place emphasis on verification of the following:
  - Information exchanges – information that flows between two operational nodes
  - Flows – sequence of events / activates involving interfaces and / or information exchanges
  - Interfaces – interconnection for two entities (e.g., systems, subsystems, user and system) that is used to exchange information

### A.1.7.5 Additional Pattern Information

The details in this section contribute to the clarity of the NCP’s intent and guidance
A.1.7.5.1 Supporting Material
Identify any supporting material, such as
- Details of prior versions
- Comments from prior reviews that were deferred for future implementation, e.g.,
  - Standards known to be in-work, or complete but not yet approved
  - New technologies that are currently not mature enough for use at this time (TRL 5 or below)\(^{22}\)
  - Due to legal implications, product liability, political considerations (e.g., inability to obtain export approval)
- Description of conflict resolution between technical factions to avoid reopening already-resolved issue(s), or discussion of conditions under which the issue(s) should be reexamined
- Names of Working Groups, Integrated Project Teams, external organizations, and individuals that worked jointly on this NCP (if applicable)
- New participants in the NCOIC are often unaware of prior discussions and often question consensus on issues—need to document the consensus process regarding NCP content.

A.1.7.5.2 Additional Implementation Guidance
This section allows NCP authors to include additional guidance as part of this appendix to the NCP where that guidance is not addressed earlier in the document. Typical NCP implementation guidance includes:
- Architecture
- Design implementation
- Requirements definition and/or validation/verification
- Procedures

Another way to understand the three categories of NCPs is by using SCOPE\(^{23}\) dimensions as follows:
- Technical NCPs represent a binding decision between one or more Net-Ready SCOPE dimensions & associated value ranges, and one or more Technical Feasibility dimensions & associated value ranges. They include/specify no SCOPE attributes in the Capability SCOPE dimension set.
- Capability NCPs include one or more binding decisions between Capability SCOPE dimensions attribute value ranges and Net-Ready SCOPE dimension attribute value ranges. They may also include binding decisions to Technical Feasibility dimension attributes and value ranges.
- Operational NCPs represent a binding decision among Capability SCOPE dimension attributes only. No Net-Ready SCOPE dimensions and associated value sets are specified, nor are any Technical Feasibility dimensions or values specified, with the possible exception of the inter-element time binding dimension pertaining to actors and actions in the Operational NCP.

A.1.7.5.3 Adherence to Relevant NCOIC Documents

\(^{22}\) Technical Readiness Levels as defined by the U.S. DoD Acquisition Guide Book, 2009
\(^{23}\) NCOIC Systems, Capabilities, Operations, Programs, and Enterprises model
This section describes the pattern's mapping to the guidance in the NIF NSD-RM and any applicable Specialized Frameworks documents. This section must include a description of how the NCP supports each principle identified.

- **NIF NSD-RM Adherence and Analysis** - Examine the NIF overarching principles, determine their applicability to the NCP under development and if applicable, verify adherence to NIF Overarching Principles (NIF NSD-RM Section 2.3.1.2).
- **Specialized Frameworks Adherence and Analysis** - Verify adherence to principles and analysis contained in Specialized Frameworks as appropriate (for example, the Net-Centric Services Framework).

### A.1.7.5.4 Validation Artifacts and/or Evidence

Provide evidence documenting what was done to validate the credibility and completeness of this NCP. Evidence may include artifacts, test results, reports, briefings, analyses, or Subject Matter Expert (SME) recommendations, from the review of this NCP.

- **Operational NCPs**: might include a description of how the NCP was derived from a validated Operational Requirements Document, a commonly accepted Concept of Operations, Operational Description (OD) or a commonly accepted Operational Scenario
- **Capability NCPs**: might include historical evidence of how this capability has been successfully used in operational missions
- **Technical NCPs**: might include description of, or test results from a prototype implementation of the technology in the NCP and field trials or experimentation

### A.1.7.5.5 Vision for the future

This section may contain a recommended roadmap for follow-up activities related to this pattern, including decision-points for updating (e.g. new/revised standards or guidance) or retiring the NCP.

### A.1.7.5.6 Formats

For actors recommend the following as appropriate.

- **UML/SysML Use Case Diagram** showing actors & use cases of the system, procedure, or operation

For structure recommend the following as appropriate.

- **UML Class Diagram**
- **SysML Block Definition Diagram** (& SysML Parametric Diagrams or Requirement Diagrams, if applicable)
- **DoDAF / MoDAF / NAF / AusDAF diagrams**: Systems & Services
- A simple Block Diagram of components

For behavior recommend the following as appropriate.
- UML / SysML Activity Diagram, Sequence Diagram (or State Machine and Timing Diagram, if applicable)
- DoDAF / MoDAF / NAF / AusDAF diagrams: Services & Capabilities
- Flow diagram of procedural steps using BPMN and associated flow diagrams, IDEF-0 or similar modeling method for Capability or Operational pattern behaviors, especially those with close coupling to commercial enterprise processes

For a Requirements Verification table, the following format is recommended:

<table>
<thead>
<tr>
<th>#</th>
<th>Title</th>
<th>Description</th>
<th>Type</th>
<th>Reference</th>
<th>Verification Method</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- A numerical tracking number that can be used to trace requirements to implementation and testing for certification purposes
- A short title to identify the requirement
- A short description of the requirement being captured
- The type of requirement (“Mandatory,” “Recommended”, or “Allowed”)
- A pointer to the applicable sections of the pattern that discuss the requirement AND any applicable sections of the underlying standards that apply. While this element must contain a linkage to the pattern guidance and applicable sections of the standards, it is by reference only and should NOT reiterate guidance verbatim from other sections of the pattern.
- List the verification methods (Analysis, Test, Demonstration, or Inspection) that are allowable for this requirement.
- List any additional information that would be useful for the Building Blocks supplier in implementing and verifying this requirement

A.2 Repository Model
A.2.1 Rationale
The NIF includes many artifacts and assets that facilitate the development of network centric systems. The NIF Repository provides a comprehensive facility to record and retrieve these assets in a uniform manner and make them available to users of the NIF. Instances of the NIF Repository can also be installed and used to record assets of individual programs, projects and systems.

The NIF Repository is based on the OMG’s RAS. Using an open standard allows the NIF Repository to be accessible from any tool that implements this standard, and provides a canonical form for the exchange of artifacts and assets among users of the NIF Repository, individual implementations or repositories based on the RAS.

A.2.1.1 Assets
Assets provide a solution to a problem for a given context. An asset is a collection of artifacts. An asset may have a variability point, which is a location in the asset that may have a value provided or customized by the asset consumer. The asset has rules for usage which are the instructions describing how the asset should be used.

A specific kind of asset may specify the artifacts that must be in the asset and may declare a specific context, such as a development context or a runtime context for which the asset is relevant. There are three key dimensions that describe reusable assets: granularity, variability, and articulation. For each framework, the NIF Framework of Frameworks (FoF) provides definitions for the following major categories of resources:
- Contextual information
- Artifacts
- Rules

A.2.2 Contextual Information
Contextual Information includes the necessary knowledge about the problem space, assumptions and other accumulated knowledge that leads to the development and use of a specific framework. Contextual Information may include definitions, dictionaries, ontologies, information models and other knowledge information. This category is also intended to capture dependency links between frameworks.

A.2.2 Architecture Description
The Architecture Description contains:
- Operational view including Use Cases and Scenarios
- Systems view including Interfaces and Patterns descriptions
- Technical view including standards and standards profile used

A.2.3 Architecture Document

The Architecture Document contains:
- Enterprise Context including
  - Relevant SCOPE dimensions
- Enterprise architecture including:
  - Business model,
  - Information model,
  - Organization model,
  - Application model
  - Performance model
  - Security model
  - ….
- OAA requirements including relevant Principles from:
  - Overarching Framework
  - Communications Framework
  - Information and Semantics Framework
  - Information Assurance Framework
  - System and Network Management Framework
- Candidate solutions assessment including:
  - selection criteria,
  - relevant patterns from above frameworks,
  - solutions benchmarks results.
- OAA design baseline is established that describe:
  - existing and planned systems
  - selected network centric patterns
  - relationships between them
- The set of systems descriptions that satisfy the requirements for the OAS are specified.
- The interface requirements are incorporated into the architectural design solution.
- The traceability of architectural design to OAS requirements is established (allocation matrix).
- A basis for verifying the system elements is defined.
- A basis for the integration of systems is established.
- A list of risks and opportunities is recorded.
- A statement of foreseeable changes (forecasts)
- A rationale of all current and obsolete design decisions is logged. This rationale will be used for changes rework including future architecture increments.
A.2.4 Repository Description

Defining, selecting and using the right assets are not easy tasks for NIF users. The role of NIF’s repository is to provide means of help for NIF users, so that they can select relevant artifacts that will guide them toward a network centric solution.

The Repository framework described below is based on the OMG RAS standard and is suggested as a Repository for storing NCOIC artifacts.

NOTE that links between the NIF and RAS models have been kept as simple as possible to enable using another Repository structure such as XML-based databases.

Figure A- 7: NIF Repository Object Model
A NIF asset has a solution that is composed of one or more artifacts. An asset is classified by a classification scheme that includes a context for the classification. An asset also is accompanied by an optional usage that includes any number of asset activities and artifact activities, where the activity of individual artifacts is used by the asset. Each of these activities can be decomposed into any number of finer grained activities. The usage is also given a context. Asset is an abstract concept. The NIF repository will define specific concrete asset types as needed. These concrete asset types are defined in the NIF core profile. Other profiles can be added to extend the NIF core profile for specific domains.

An artifact is a work product that can be created, stored and manipulated by asset producers, consumers and by tools. An artifact is either an actual file located in the asset’s package, or represents a logical entity that contains at least one child artifact that is an actual file. An artifact can be decomposed into any number of finer grained artifacts. They are further defined by an artifact type. A dependency graph can be constructed in addition to the decomposition for the artifacts. With ever increasing needs for security, an artifact may be encrypted with a specific algorithm. The digest-name and digest-value attributes contain the name and value of such encryption activities. An artifact also is assigned a context. A context may be categorized for example as core, business, development, etc.

Asset, artifact, context and activity are named elements that include an optional description. The repository entries can also be characterized by a profile and any number of related profiles that are also named elements. The profiles and assets can include versions. See the OMG Reusable Asset Specification: [http://www.omg.org/technology/documents/formal/ras.htm](http://www.omg.org/technology/documents/formal/ras.htm)
B ANNEX B – “Don’ts” and Anti-patterns

<To be included in a future version of the NIF>
C  ANNEX C – Modeling

C.1  Metamodeling Framework - Introduction

This annex describes a model to guide the use and development of network centric architectures for NCOIC stakeholders by defining specific types of information artifacts such as patterns, describing each information artifact’s role in the model, and identifying the type of relationships between the information artifacts.

The model’s purpose is not to represent NCOIC’s organizational structure and its working groups, nor to represent the relationships between specific documents or deliverables created by the NCOIC, but rather to represent the types of information relevant to the purpose of architecture and the necessary relationships that hold between the operational and technical aspects described in various types of architectural information artifacts.

C.2  NIF Model Goals

The NIF information artifact model has the following purposes:

1. Architecture development - provide guidance to create architecting assets. Identify the information elements and the dependency relationships between these elements, within an architecture template and model
2. Navigation and discovery of architecting assets - define a storage model for architecting assets that identifies the navigation paths for finding information relevant to a given asset, e.g., "Which patterns support a specific Network Centric Operations Capability?", "Which patterns mitigate a Network Centric Interoperability problem?". This model should serve as a basis for creating tools that support the NCOIC process.

The NSD-RM encourages the use a storage model that is compliant with the OMG/RAS standard; however, other storage models can also be used, provided they are sufficiently open to host NIF architecting assets.

C.3  Methodology Strategy

The NIF methodology combines top-down, middle-out and bottom-up architectural approaches:

- Top-down analyzes net-centricity starting from operational needs
- Middle-out introduces a priori solution elements based on OAA patterns
- Bottom-up builds on existing candidate solutions based on specialized patterns.
The NIF recommends that these three approaches be applied successively, in several cycles where each cycle refines the breadth and depth of previous results.

Depending on the maturity of the domain and project, specific architectural approaches should be applied
- Newer domains require stricter top-down analysis
- Well-established domains and projects need more cursory top-down and middle-out analyses before a bottom-up analysis

Additionally, the NIF methodology recommends decomposing each operational, architectural and technical view using a network-oriented paradigm such as “nodes” and “edges”. This approach is compliant with other architectural frameworks such as the DoDAF.

C.4 Overview

The model is divided in 2 areas:
1. A conceptual metamodel that positions architecture concepts within the enterprise environment
2. An architecture framework that exposes operations, systems and technical relationships - the model specifies the operations, system and technical information elements to be used in architecture descriptions

C.4.1 Architecture within Enterprise Context

![Architecture Decomposition Diagram]

Figure C-1: Architecture Decomposition
C.4.2 Operational, System and Technical Descriptions

The NIF information artifact model covers the operational, system and technical views and their relationships through operational, system and technical descriptions. Depending on the view, the network-oriented paradigm is applied as follows:

- In the operational view, nodes are operational assets and edges are determined by need-lines
- In the system view, nodes are systems and edges are determined by associations
- In the technical view, nodes are patterns or standards and edges are determined by interfaces

The operational view drives the construction of the system view which, in turn, drives the construction of the technical view.

Prior to developing an operational description (OD), the NIF recommends describing a full enterprise architecture that will explicitly describe the primary enterprise models—those such as business, information, process and security models. Regardless of recommendations that result from adhering to its principles, NIF does not define or suggest a particular format to describe an enterprise architecture, as this is beyond NCOIC’s mission.
The OD captures all relevant operational information that will support the development of an OAA and patterns. The OD is primarily based on network centric dimensions, scenarios, use cases, capabilities and problems (gaps). These artifacts are described below:

- Dimensions capture the network centric context and result from a SCOPE analysis. SCOPE dimensions help to state global network centric objectives and avoid narrowing the network centric capabilities to specific scenarios.
- NCO scenarios capture broad representative mission threads that involve network centric operations. NCO scenarios result from operational analysis that is jointly led by architects and subject matter experts (SME).
- NCO use case describes a subset of operations that focus on a scenario branch.
- NCO capabilities identify those required to support NCO scenarios. NCO capabilities are identified when refining the NCO scenarios.
- NCO problems (including gaps) summarize those challenges to overcome. NCO problems will help to identify relevant technical solutions.

In parallel, general network centric security and safety requirements must be stated.

The main difficulties here are 1) to extract NC requirements from general requirements and 2) to ensure that all NC requirements have been captured. NC requirements can usually be extracted from general requirements by abstracting them via modeling techniques such as “Entity/Relationship”, UML, or other. Checking for completion of NC requirements can be achieved through successive reviews and is confirmed when reviews identify no additional NC requirements.

In addition to the list of primary OD artifacts listed above, and during operational analysis, ODs may usefully identify then capture operational patterns that are patterns whose participating elements are operational assets.

The architectural system description targets to specify the Overarching Architecture (OAA) and comprise:

- Identify then apply relevant OAF Principles (note that OAF principles apply to functional, logical and physical architecture levels),
- Identify required global NC functions and services that will support NCO capabilities,
- Identify and describe system-level functions and services that support NC functions/services
  - Consider Network Centric Enterprise Integration capabilities (NCEI) (e.g., communications, integration, information, information assurance, and system and network management).
- Identify conceptual, logical and physical information model
  - Conceptual Models the user concepts in terms familiar to users
  - Logical More formal model that considers unique data representation, emphasis on well-defined semantics and exclusivity, and domain-level completeness
  - Physical Models all the information necessary for database implementation
- Identify and (externally) describe participating systems’ architectures (e.g., interfaces).
  Participating systems may include existing, modernized and green-field systems,
- Allocate NC functions/services to participating systems,
- Identify gaps and candidate solutions including existing patterns,
- Plan OAA evolution over time (Architecture Maturity Model) using forecasts
- Record architecture design decisions and risks/opportunities

Note: OAA’s main role is to ensure consistency in global network centric design.

The main difficulties are likely to be collecting relevant OA architecting requirements and assessing tradeoffs (analysis of alternatives).

The technical description supports refining the OAA Architecture NCEI Capabilities (e.g., communication, information, integration, information assurance, and system and network management). Network centric technical information artifacts comprise NC technical capabilities, NC technical problems, NC technical patterns, NC technical tenets, NC technical standards and standards profiles. NIF does not require a specific format for these information artifacts. Various methodologies and formats are available as proposed by organization such as IEEE and DISA.

  - NCT capabilities express technical capabilities that will support NC functions/services,
  - NCT problems express technical problems to be solved to achieve NCT capabilities,
  - NCT tenets express technical tenets to apply for solving NCT Problems,
  - NCT patterns express technical patterns that provide ready-to-use solutions for (part of) NCT problems.

The main challenge is likely to be identifying and then describing new relevant patterns. This generally requires several identification-design-test cycles: at least three according to the literature, but probably more cycles, if a pattern is complex or addresses a new domain.

**NIF Model Goal**

The NIF Information Artifact Model has two main purposes:

1. **Architecture development** - provide guidance to create architecting assets - identify the information elements and the dependency relationships between these elements within an architecture template and model

2. **Architecting assets navigation and discovery** - define a storage model that identifies the paths for finding information relevant to a given asset. For example, "which patterns support a specific network centric operations capability?" or "which patterns mitigate a network centric interoperability problem?"

The NSD-RM suggests using a storage model that is compliant with the OMG/RAS standard, but another model can be used if it is “open” enough to host NIF’s architecting assets.
**NCOIC Information Artifact Model Tenets**
These tenets are required to enable the pattern goals stated above:

1. Conceptual metamodel - the NCOIC Information Artifact diagram represents a conceptual metamodel specifying the information artifacts related to development of NCOIC architecting artifacts and the relationships between them.
2. Operations and technical relationships - the model specifies both operations and technical information elements to be used in architecture descriptions.

### C.4.2.1 Operational Description

![Diagram of operational description with NcoContext, NcoMission, NcoScenario, NcoUseCase, NcoCapability, NcoAsset, NcoNeedline, NcoPattern, and NcoProblem relationships.]

Figure C-3: Operational Description
<table>
<thead>
<tr>
<th>Model Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCO Context</td>
<td>Describes global NCO context and actors. Context described using NCOIC/ SCOPE analysis.</td>
</tr>
<tr>
<td>NCO Mission</td>
<td>Describes NCO mission.</td>
</tr>
<tr>
<td>NCO Scenario</td>
<td>Describes NCO scenarios that break down the mission into its threads.</td>
</tr>
<tr>
<td>NCO UseCase</td>
<td>Describes NCO use cases that refine scenarios and help to identify NCO capabilities.</td>
</tr>
<tr>
<td>NCO Capability</td>
<td>Describes NCO capabilities that support use cases.</td>
</tr>
<tr>
<td>NCO Problem</td>
<td>Describes NCO problems to be solved in order to fulfill capabilities.</td>
</tr>
</tbody>
</table>
| NCO Asset       | Describes NCO assets used in mission, scenarios and their use cases.  
Assets categories map to doctrine, organization, training, materiel, leadership, personnel, and facilities (DOTMLPF) areas. Assets include data, data model and metadata. |
| NCO Needline    | Describes NCO need-lines that convey exchanges between NCO assets and actors.  
A need-line is a requirement that is the logical expression of the need to transfer, for example, information among nodes.                                                                                   |
| NCO Pattern     | Describes NCO pattern that captures a configuration of the following elements:  
- NCO problem  
- NCO context  
- NCO capability  
- NCO need-line |
C.4.2.2 System Architecture Description

Figure C-4 illustrates the main architecting artifacts (reuse parts of IEEE 1470).

![System Architecture Diagram]

**Figure C-4: System Architecture Description**

<table>
<thead>
<tr>
<th>Model Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NifOverarchingArchitecture</td>
<td>Shows Overarching Architecture (OAA).</td>
</tr>
<tr>
<td>NifSystem</td>
<td>Shows any participating system (or SoS) within OAA.</td>
</tr>
<tr>
<td>NcaSystemNode</td>
<td>Shows a system’s node when system is distributed across multiple operational nodes.</td>
</tr>
<tr>
<td>NifSystemRelationships</td>
<td>Shows any relationships between two participating systems.</td>
</tr>
<tr>
<td>NifInformationModel</td>
<td>Shows global information model (ex: JC3IEDM).</td>
</tr>
</tbody>
</table>
NifInterface | Shows any interface between two participating systems.
---|---
NifService | Shows any function or service.
NifQoS | Shows Quality of Service that splits into:
- QoOS : Quality of Operational Services
- QoIS : Quality of Information Services
- QoTS : Quality of Transmission Services
NctPattern | Contains a technical pattern that supports a service.
NifStakeholder | Shows any architecture stakeholder
NifForecasts | Contains architecture forecasts (NOT technology ones)
NifRisksOpportunities | Shows architecture risks and opportunities
NifDesignDecisions | Shows a history log that captures design decisions

C.4.2.3 Technical Description

The technical description is centered on the technical pattern artifact and shown in Figure C- 5.
Figure C-5: Technical Description

Table C-3: Technical Description Elements

<table>
<thead>
<tr>
<th>Model Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NctContext</td>
<td>Describes a network centric technical context.</td>
</tr>
<tr>
<td>NctCapability</td>
<td>Describes a network centric technical capability.</td>
</tr>
<tr>
<td>NctProblem</td>
<td>Describes a network centric problem to be solved.</td>
</tr>
<tr>
<td>NctStandard</td>
<td>Describes a network centric standard to be used.</td>
</tr>
<tr>
<td>NctStandardProfile</td>
<td>Describes a set of network centric standards to be used.</td>
</tr>
</tbody>
</table>
NctPrinciple States a network centric principle that applies.
NctTenet States a network centric tenet that applies.

C.4.2.4 NIF Framework

Figure C-6 illustrates the main relationships for NifFramework. NifFramework describes the NIF Framework concept.

Figure C-6: NIF Framework Relationships
### Table C-4: NIF Framework Relationship Elements

<table>
<thead>
<tr>
<th>Model Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NifStakeholder</td>
<td>Figure any NIF Stakeholder</td>
</tr>
<tr>
<td>NifUser</td>
<td>Figure any NIF User that operates the framework</td>
</tr>
<tr>
<td>Standards</td>
<td>Each NifFramework gathers the set of relevant standards.</td>
</tr>
<tr>
<td>NifFramework</td>
<td>Figure any NIF Architecting Framework. A framework reuses NifFmwkResources to produce NifFmwkProducts under the control of NifUser. Note that a NifFramework may contain another NifFrameworks.</td>
</tr>
<tr>
<td>NifOverArchingFramework</td>
<td>Figure the Overarching Framework.</td>
</tr>
<tr>
<td>NifSpecializedFramework</td>
<td>Figure any Specialized Framework</td>
</tr>
<tr>
<td>NifResource</td>
<td>Figure any framework resource.</td>
</tr>
<tr>
<td>NifConcept</td>
<td>Figure any framework concept. May be linked with any other framework concept.</td>
</tr>
<tr>
<td>NifProduct</td>
<td>Figure any framework product including OD, view, service, pattern, PFC, profile …</td>
</tr>
<tr>
<td>NifPrinciple</td>
<td>Figure any framework Principle. Principle constrains NifProcess and NifFmwkProducts. May be linked with any other framework Principle.</td>
</tr>
<tr>
<td>NifFmwkProduct</td>
<td>Figure any NifFmwkProduct produced with NifFramework</td>
</tr>
<tr>
<td>NifProcess</td>
<td>Figure any framework process. May be linked with any other framework process..</td>
</tr>
<tr>
<td>NifRelationship</td>
<td>Figure any relationship between two architecture components.</td>
</tr>
<tr>
<td>NctPattern</td>
<td>Figure a technical pattern Note that Pattern may be both a product and a resource.</td>
</tr>
<tr>
<td>NctPFC</td>
<td>Figure a pattern exclusively using protocols (Pattern For Communications).</td>
</tr>
<tr>
<td>NctService</td>
<td>Figure a service.</td>
</tr>
</tbody>
</table>
C.4.2.5 RAS “Core” Model

Figure C-7 illustrates the core elements of the Reusable Asset Specification (RAS) model which could form the basis for a NIF repository.

This model shown in Figure C-7 is an extract from OMG RAS Model. The full RAS model is described within the OMG RAS standard that can be accessed on OMG’s website (www.omg.org).

Figure C-7: RAS Core Model
C.4.2.6 NIF/RAS Mapping Model

Figure C-8 shows how NIF artifacts can be integrated into the RAS storage model by identifying the links between core RAS elements and NIF elements. In fact, NIF elements are simply specialized implementations of the `RasNamedAssetElement` element.

![Diagram of NIF/RAS Model Mapping](image)
C.4.2.7  NIF/CADM Mapping Model

Table C-5 shows the mapping between NIF and DoDAF/Core Architecture Data Model (CADM) v1.5 views.

<table>
<thead>
<tr>
<th>Recommended NIF View</th>
<th>Contents</th>
<th>DoDAF View</th>
</tr>
</thead>
<tbody>
<tr>
<td>NCAV-2</td>
<td>Integrated Data Dictionary</td>
<td>AV-2</td>
</tr>
<tr>
<td>NCOV-2</td>
<td>Operational Connectivity Description</td>
<td>OV-2</td>
</tr>
<tr>
<td>NCOV-3</td>
<td>Operational Information Exchange Matrix</td>
<td>OV-3</td>
</tr>
<tr>
<td>NCOV-7</td>
<td>Logical Data Model</td>
<td>OV-7</td>
</tr>
<tr>
<td>NCSV-1</td>
<td>Systems Interface Description</td>
<td>SV-1</td>
</tr>
<tr>
<td>NCSV-2</td>
<td>Systems Communication Description</td>
<td>SV-2</td>
</tr>
<tr>
<td>NCSV-3</td>
<td>NCSV-3 System-to-Systems Matrix</td>
<td>SV-3</td>
</tr>
<tr>
<td>NCSV-4</td>
<td>Systems Functionality Description</td>
<td>SV-4</td>
</tr>
<tr>
<td>NCSV-6</td>
<td>System Data Exchange Matrix</td>
<td>SV-6</td>
</tr>
<tr>
<td>NCSV-7</td>
<td>Systems Performance Parameters Matrix</td>
<td>SV-7</td>
</tr>
<tr>
<td>NCSV-8</td>
<td>Systems Evolution Description</td>
<td>SV-8</td>
</tr>
<tr>
<td>NCSV-9</td>
<td>Systems Technology Forecast</td>
<td>SV-9</td>
</tr>
<tr>
<td>NCSV-11</td>
<td>Physical Schema</td>
<td>SV-11</td>
</tr>
<tr>
<td>NCTV-1</td>
<td>Technical Standards Profile</td>
<td>TV-1</td>
</tr>
<tr>
<td>NCTV-2</td>
<td>Technical Standards Forecasts</td>
<td>TV-2</td>
</tr>
</tbody>
</table>

C.4.2.8  NIF/PFM Mapping Model

<Placeholder to describe mapping between NIF and NCOIC/SF FT/PFM (Pattern Focused Model) in a future version of the NSD-RM>
## Annex D – Technical Readiness (TRL) Scale

### Table D-1: Technical Readiness Levels

<table>
<thead>
<tr>
<th>Technology Readiness Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Basic principles observed and reported</td>
<td>Lowest level of technology readiness. Scientific research begins to be translated into applied research and development. Examples might include paper studies of a technology’s basic properties.</td>
</tr>
<tr>
<td>2. Technology concept and/or application formulated</td>
<td>Invention begins. Once basic principles are observed, practical applications can be invented. Applications are speculative and there may be no proof or detailed analysis to support the assumptions. Examples are limited to analytic studies.</td>
</tr>
<tr>
<td>3. Analytical and experimental critical function and/or characteristic proof of concept</td>
<td>Active research and development is initiated. This includes analytical studies and laboratory studies to physically validate analytical predictions of separate elements of the technology. Examples include components that are not yet integrated or representative.</td>
</tr>
<tr>
<td>4. Component and/or breadboard validation in laboratory environment</td>
<td>Basic technological components are integrated to establish that they will work together. This is relatively “low fidelity” compared to the eventual system. Examples include integration of “ad hoc” hardware in the laboratory.</td>
</tr>
<tr>
<td>5. Component and/or breadboard validation in relevant environment</td>
<td>Fidelity of breadboard technology increases significantly. The basic technological components are integrated with reasonably realistic supporting elements so it can be tested in a simulated environment. Examples include “high-fidelity” laboratory integration of components.</td>
</tr>
<tr>
<td>6. System/subsystem model or prototype demonstration in a relevant environment</td>
<td>Representative model or prototype system, which is well beyond that of TRL 5, is tested in a relevant environment. Represents a major step in a technology’s demonstrated readiness. Examples include testing a prototype in a high-fidelity laboratory environment or in a simulated operational environment.</td>
</tr>
<tr>
<td>7. System prototype demonstration in an operational environment</td>
<td>Prototype near, or at, planned operational system. Represents a major step up from TRL 6, requiring demonstration of an actual system prototype in an operational environment such as an aircraft, vehicle, or space. Examples include testing the prototype in a test bed aircraft.</td>
</tr>
<tr>
<td>Technology Readiness Level</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>8. Actual system completed and qualified through test and demonstration</strong></td>
<td>Technology has been proven to work in its final form and under expected conditions. In almost all cases, this TRL represents the end of true system development. Examples include developmental test and evaluation of the system in its intended weapon system to determine if it meets design specifications.</td>
</tr>
<tr>
<td><strong>9. Actual system proven through successful mission operations</strong></td>
<td>Actual application of the technology in its final form and under mission conditions, such as those encountered in operational test and evaluation. Examples include using the system under operational mission conditions.</td>
</tr>
</tbody>
</table>
E  ANNEX E – Changes Log and Open Issues

The following template is recommended for reporting comments about this document to the NCOIC.

Table E-1: NSD-RM Change Log and Open Issues

<table>
<thead>
<tr>
<th>Comment Tracking #</th>
<th>Log-in Date</th>
<th>Section/Page</th>
<th>Comment Recommendation</th>
<th>Disposition (accepted, completed, deferred, rejected)</th>
<th>Resolution / Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
F    Annex F – References

The following documents are referenced in the NSD-RM. They are also listed here for a reader’s convenience.


12. NCOIC. “Network Centric Assessment Tool (NCAT™).”

13. NCOIC. “NIF Scope and Problem Statement (NSPS) version 0.9,” February 27, 2007.

14. NCOIC. “Systems, Capabilities, Operations, Programs, and Enterprises (SCOPE) Model.”


