NCIA Registry 0 4 FEB 2015 The Hague



NATO Communications and Information Agency Agence OTAN d'information et de communication

AGENCY INSTRUCTION INSTR TECH 06.02.11 SERVICE INTERFACE PROFILE FOR A NOTIFICATION CACHE SERVICE

Effective date:

Issued by:

Revision No:

Chief, Core Enterprise Services

Original

18 MAR Shaurion

Approved by:

Director Service Strategy _

NATO UNCLASSIFIED



Table of Amendments

Amendment No	Date issued	Remarks

Author Details

Organization	Name	Contact Email/Phone	
NCI Agency	A. Gruler	alexander.gruler@ncia.nato.int	
NCI Agency	V. De Sortis	vincenzo.de.sortis@ncia.nato.int	



Table of Contents

_					_
С) /	٩.	c	1	-
г		٦	G	11	-

0	PRELIMINARY INFORMATION	4
0.1 0.2 0.3	References Purpose Applicability	4 4 4
1	SIP INTRODUCTION	4
1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 1.10	Audience	5 5 6 6 7 7 7 7
2	BACKGROUND	8
2.1 2.2 2.3	Operational Requirements Service Description Typical Service Interactions	8 8 8
3	SERVICE INTERFACE SPECIFICATION	9
3.1 3.2 3.3 3.4	Interface Overview Technical Requirements Operations Behaviours Not Specified in the Notification Cache	9 9 10 13
4	REFERENCES	14
5	ABBREVIATIONS	16

List of Annexes

ANNEX 1 - NON-NORMATIVE INTERFACE SPECIFICATION	ANNEX 1 - NO	N-NORMATIVE INTERFACE SPECIFICATION		17
---	--------------	-------------------------------------	--	----





AGENCY INSTRUCTION 06.02.11

SERVICE INTERFACE PROFILE FOR A NOTIFICATION CACHE SERVICE

0 PRELIMINARY INFORMATION

0.1 References

- A. NCIA/GM/2012/235; Directive 1 Revision 1; dated 3 May 2013
- B. NCIARECCEN-4-22852 DIRECTIVE 01.01, Agency Policy on Management and Control of Directives, Notices, Processes, Procedures and Instructions, dated 20 May 2014
- C. NCIARECCEN-4-23297, Directive 06.00.01, Management and Control of Directives, Processes, Procedures and Instructions on Service Management, dated 03 June 2014

0.2 Purpose

This Technical Instruction (TI) provides detailed information, guidance, instructions, standards and criteria to be used when planning, programming, and designing Agency products and services. In this specific case the TI defines a Service Interface Profile (SIP) for one NATO's Core Enterprise Services.

TIs are living documents and will be periodically reviewed, updated, and made available to Agency staff as part of the Service Strategy responsibility as Design Authority. Technical content of these instructions is the shared responsibility of SStrat/Service Engineering and Architecture Branch and the Service Line of the discipline involved.

TIs are primarily disseminated electronically¹, and will be announced through Agency Routine Orders. Hard copies or local electronic copies should be checked against the current electronic version prior to use to assure that the latest instructions are used.

0.3 Applicability

This TI applies to all elements of the Agency, in particular to all NCI Agency staff involved in development of IT services or software products. It is the responsibility of all NCI Agency Programme, Service, Product and Project Managers to ensure the implementation of this technical instruction and to incorporate its content into relevant contractual documentation for external suppliers.

1 SIP INTRODUCTION

One of the main concepts of the future NATO Network Enabled Capabilities (NNEC) is that of a "network of networks"; that is, instead of a single, all-encompassing global network the NNEC environment will be made up of many NATO and national networks linked together. In order to ensure compatibility between services running in this environment there is a need for a standard (and standards-based) profile, mandatory for all service operations.

This Service Interface Profile (SIP) describes the key elements that make up the NNEC Core Enterprise Services (CES) *Notification Cache* service. It describes and profiles the operations which a *Notification Cache* service offers together with the associated message formats, and serves as a template and guideline for implementations.

A Notification Cache service offers optional functionality for publish/subscribe-based messaging. Extending the functionality of a classical WS-Notification publish/subscribe scenario (see [OASIS

¹ https://servicestrategy.nr.ncia/SitePages/Agency%20Directives%20(Technical).aspx



WSN, 2006]), the *Notification Cache* service covers the requirement to cache *Notification Messages*, and to provide retrospective access to them ([NC3A RD-3292, 2011]).

We describe the operations together with the corresponding data messages that a *Notification Cache* service offers through its interface in Chapter 2. The remainder of this chapter sets the context for this SIP, defines the naming and notational conventions used throughout the document, and lists associated standards and specifications.

1.1 Audience

The target audience for this specification is the broad community of NNEC stakeholders, who are delivering capability in an NNEC environment, or anticipate that their services may be used in this environment.

These may include (but are not limited to):

- Project Managers procuring NATO communication and information systems.
- The architects and developers of service consumers and providers that interact with the *Notification Cache Service SIP Proposal.*
- Coalition partners whose services may need to interact with NNEC services.
- System integrators delivering systems into the NATO environment.

1.2 Notational Conventions

The following notational conventions apply to this document:

- The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [IETF RFC 2119, 1997].
- Words in italics indicate references to terms defined in Section 1.3.
- Courier font indicates syntax derived from the family of WS-Notification standards in [OASIS WSN, 2006] and other open standards such as [W3C WS-Addressing Core, 2006], [W3C XPath 3.0, 2011], [W3C XML Namespaces 1.1, 2006]. In addition, Courier font is also used for identifiers such as interface names, operation names, and message names that appear in web services description languages (WSDL) and extensible markup language (XML) Schema definitions.
- Where pseudo-schemas are provided for a component, they use BNF (Backus-Naur Form) style conventions for attributes and elements. Individual symbols have the following meaning:
 - o '?' denotes optionality (i.e. zero or one occurrence).
 - o '*' denotes zero or more occurrences.
 - o '+' denotes one or more occurrences.
 - o '[' and ']' are used to form groups.
 - o '|' expresses choice.

1.3 Taxonomy Allocation

This service falls under the following allocation under the C3 Taxonomy [NATO TIDE Community, 2012]:

Technical Services \rightarrow Core Enterprise Services \rightarrow SOA Platform Services \rightarrow Message-Oriented Middleware Services



1.4 Terminology

This document shares terminology with the "Notification Cache Technical Specification" [NCIA TR/2012/SPW008423/10, 2014]. Please refer to this document for the latest terminology.

1.5 Namespaces

The following namespaces are used in this document:

Prefix	Namespace	
ncs	http://ncia.nato.int/ces/notificationcacheservice_1_1_0/service	
ncs-msg	http://ncia.nato.int/ces/notificationcacheservice_1_1_0/message	
soap	http://schemas.xmlsoap.org/soap/envelope/ or http://www.w3.org/2003/05/soap-envelope2	
wsa	http://www.w3.org/2005/08/addressing	
wsn	http://docs.oasis-open.org/wsn/b-2	
wsnt	http://docs.oasis-open.org/wsn/b-2	
wsn-br	http://docs.oasis-open.org/wsn/br-2	
wsn-bw	http://docs.oasis-open.org/wsn/bw-2	
wsrf-bf	http://docs.oasis-open.org/wsrf/bf-2	
xs	http://www.w3.org/2001/XMLSchema	
xsd	http://www.w3.org/2001/XMLSchema	

1.6 Goals

This Service Interface Profile is intended to give directives, along with clarifications and amendments, on the use of mandatory and recommended interfaces and data structures to be implemented by the components of the NNEC CES that offer a *Notification Cache* service.

The following topics are in the scope of this profile:

- Definition of the referenced normative standards
- Definition of the optional and mandatory interfaces for the Notification Cache service
- Definition of the optional and mandatory operation for the interfaces
- Definition of the non-standardized behaviours.

² See [NC3A RD-3143, 2011].



1.7 Non-Goals

The following topics are outside the scope of this profile:

- Definition of the format of Notification payloads.
- Definition of specific *Notification* filter languages.
- Definition of mechanisms and/or concepts by which a *Notification Cache* can be discovered by *Notification Requestors* or *Notification Producers*.
- Definition of security mechanisms and how they can be applied to the message exchange with the *Notification Cache*.
- Definition of non-functional messaging characteristics such as reliability, message compression etc.

1.8 Relationships to other Services

This SIP is directly related with Notification Cache Services (as described in [NCIA TR/2012/SPW008423/10, 2014]) and indirectly related to the WS-Notification publish/subscribe services (as described in [OASIS WSN, 2006]). Both the services are described under the C3 Taxonomy [NATO TIDE Community, 2012] as:

 Technical Services | Core Enterprise Services | SOA Platform Services | Message-oriented Middleware Services

1.9 Relationships to Other Profiles and Specifications

1.9.1 OASIS Web Services Notification (WSN) Technical Committee (TC)

WS-Notification [OASIS WSN, 2006] is a family of related specifications that define a standard web services based approach to notification using a topic-based publish/subscribe messaging pattern. It includes standard message exchanges to be implemented by service providers that wish to participate in:

- Notification-based message exchange
- Standard message exchanges for a *Notification Broker* service provider (allowing publication of messages from entities that are not themselves service providers)
- Operational requirements expected of service providers and requestors that participate in notification-based message exchange
- An XML model that describes *Topics*.

1.9.2 Relevant NATO CES documents

This specification should be read in conjunction with the following documents which provide more detailed technical information about the publish/subscribe architectures, the WS-Notification specification and the Notification Cache Service specification:

- * "Notification Cache Technical Specification" [NCIA TR/2012/SPW008423/10, 2014]
- "Publish/Subscribe Service Interface Profile Proposal. Version 1.0" [NC3A RD-3139, 2011].

1.10 Normative References

The following documents have fed into this specification, and are incorporated as normative references:

- WS Base Notification Standard 1.3 [OASIS WS-BaseNotification 1.3, 2006], on-line at http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf
- WS Brokered Notification Standard 1.3 [OASIS WS-BrokeredNotification 1.3, 2006], on-line at http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-os.pdf



- WS Base Faults Standard 1.2. [OASIS WS-BaseFaults 1.2, 2006], on-line at http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf
- WS Addressing Standard [W3C WS-Addressing Core, 2006], on-line at http://www.w3.org/TR/2006/REC-ws-addr-core-20060509

2 BACKGROUND

2.1 Operational Requirements

OASIS WS-Notification [OASIS WSN, 2006] publish/subscribe architectures offer a selective dissemination scheme that delivers the published content only to the interested receivers. Like most of the common publish/subscribe architectures it does not offer retroactive access to messages that were published prior to the start of a subscription or that were not received or not processed by the receiver.

One of the common requirements for the message receivers is to have access to previously published information.

Note that this requirement cannot be fulfilled with the re-publication of that information because:

- The data producer/broker may not have a copy of the information to resend.
- As result of re-publishing the information, all the subscribed consumers will receive it irrespective of them having already received it or not in the past. In this case some consumers will receive the same information via duplicated messages.

2.2 Service Description

The main purpose of a *Notification Cache* service is to act as storage for *Notification Messages* and to make them available to other services for future retrieval. The caching functionality extends a typical publish/subscribe scenario, where *Notifications* are usually no longer available after they have been published to the subscribed *Consumers*.

2.3 Typical Service Interactions

In order to retrieve cached *Notifications*, a *Notifications Requestor* has to send a request to the *Notification Cache* by means of a GetNotifications operation. The operation can be used in two different ways:

- Either in a direct request/response style (referred to as *Active Synchronization*), where a *Notification Requestor* characterizes *Notification Messages* which it wants to retrieve, and gets the appropriate *Notification Messages* directly back in the response, or
- In an indirect style (referred to as *Passive Synchronization*), where a *Notification Requestor* sends a request specifying the *Notifications* that it wants to retrieve, and additionally specifying the address of the *Consumer* to which the *Cache* should send the selected *Notifications* after retrieval.

Note that this is just a brief overview of the functionality. A much more detailed description of the functionality provided by a *Notification Cache* service can be found in [NCIA TR/2012/SPW008423/10, 2014].



3 SERVICE INTERFACE SPECIFICATION

3.1 Interface Overview

The Notification Cache service functionalities are exposed as a set of interfaces which define the message exchange patterns and the interaction between the Notification Cache and the Notification Producers, Notification Consumers, and Notification Requestors.

The *Notification Cache* may be implemented as a stand-alone service or may be embedded in other services like *Notification Producer* or *Notification Broker*. Depending on the implementation style the way in which the *Notifications* are fed inside the *Notification Cache* and the related interface configuration may vary.

In the following NON-NORMATIVE table are listed the interfaces and the related operations implemented by the *Notification Cache* service with the indication whether they are mandatory, recommended or optional in relation to the *Notification Cache* implementation style.

Interface	Operation	Stand-alone implementation	Embedded implementation
RetrieveNotification	GetNotification	Mandatory	Mandatory.
NotificationConsumer	Notify	Mandatory	Mandatory, but the one offered by the embedding service MUST be used.
CreatePullPoint	CreatePullPoint	Recommended	Recommended, but offered by the embedding service.
PullPoint	DestroyPullPoint	Recommended	Recommended, but offered by the embedding service.
PullPoint	GetMessages	Recommended	Recommended, but offered by the embedding service.

The Notification *Cache* interfaces are specified and described in detail in "Notification Cache Technical Specification" [NCIA TR/2012/SPW008423/10, 2014].

3.2 Technical Requirements

A *Notification Cache* service MUST comply to the requirements specified in [NCIA TR/2012/SPW008423/10, 2014] unless stated differently in this document. In order to avoid unnecessary repetition in this document we only profile the main concepts, the refinements and the changes compared to [NCIA TR/2012/SPW008423/10, 2014].

The Notification Cache is at the same time a consumer and producer for the Notifications. During the caching phase it receives and stores Notifications Messages acting as Notification Consumer. During a Passive Synchronization, sending Notifications to Notification Consumers, it acts as Notification Producer.

When acting as a Notification Consumer:

DECLASSIFIED - PUBLICLY DISCLOSED - PDN(2015)0018 - DÉCLASSIFIÉ - MIS EN LECTURE PUBLIQUE



- If embedded in other services (NON stand-alone) a *Notification Cache* MUST NOT expose a dedicated NotificationConsumer interface. In this case the caching functionality MUST refer to *Notifications* which are exchanged as part of an existing operation of the component (*Notification Broker, Producer, Consumer*) in which the *Cache* is embedded.
- If implemented as a stand-alone service a Notification Cache MUST support the NotificationConsumer interface.

The following requirements apply to the Notification Cache interfaces.

- A Notification Cache MUST support the RetrieveNotification interface as specified in Section 3.2 of [NCIA TR/2012/SPW008423/10, 2014]. The REQUIRED operation to implement is:
 - GetNotifications To query the Notification Cache and to retrieve selected Notifications.
- If the *Notification Cache* exposes a dedicated NotificationConsumer interface, it MUST be in line with the following specifications:
 - o WS-Notification [OASIS WSN, 2006]
 - o Section 2.2 and Chapter 4 of the [NCIA TR/2012/SPW008423/10, 2014]
 - o Annex A of [NC3A RD-3139, 2011].

The REQUIRED operation to implement is:

- o Notify To explicitly insert *Notifications* into the *Cache*.
- If embedded in other services (NON stand-alone) a Notification Cache SHOULD support the Pull-Style Notification but MUST NOT expose dedicated interfaces relying, instead, on the ones offered by the embedding service if any.
- If implemented as a stand-alone service a *Notification Cache* SHOULD support the *Pull-Style Notification*.
 - If the Notification Cache offers the Pull-Style Notification it MUST offer the CreatePullPoint interface as specified in Section 2.2.2 of [NCIA TR/2012/SPW008423/10, 2014]. The REQUIRED operations to implement are:
 - CreatePullPoint To create a new Pull Point.
 - If the Notification Cache offers the Pull Style Notification, it SHOULD offer the PullPoint interface as specified in Section 2.2.2 of [NCIA TR/2012/SPW008423/10, 2014]. The REQUIRED operations to implement are:
 - DestroyPullPoint to destroy a Pull Point
 - GetMessages to return the Notifications published to a Pull Point.

3.3 Operations

3.3.1 Operation GetNotifications

The Notification Cache MUST implement the RetrieveNotification interface as specified in Chapter 3 of [NCIA TR/2012/SPW008423/10, 2014], and MUST support the GetNotifications message exchange. A Notification Requestor MUST use the GetNotifications operation to request Notifications from the Notification Cache for both the Active Synchronization and the Passive Synchronization scenarios.

No additional rules or limitations are defined other than those specified in the corresponding WSN standard specifications of [OASIS WS-Base Notification 1.3, 2006], [OASIS WS-BrokeredNotification 1.3, 2006].



3.3.1.1 Data types

See Chapter 3 of [NCIA TR/2012/SPW008423/10, 2014].

3.3.1.2 Inputs

See Section 3.2.1 of [NCIA TR/2012/SPW008423/10, 2014].

3.3.1.3 Outputs

See Section 3.2.2 of [NCIA TR/2012/SPW008423/10, 2014].

3.3.1.4 Errors and faults

See Section 3.2.3 of [NCIA TR/2012/SPW008423/10, 2014].

3.3.2 Operation Notify

When the NotificationConsumer interface is implemented, the Notification Cache MUST support the wsnt:Notify message exchange format ([OASIS WS-BaseNotification 1.3, 2006]), and MAY support raw Notifications for receive and send Notification Messages. A Publisher MUST use the wsn-bw:Notify operation to feed new Notification Messages to the Notification Cache (in the case where the Cache is implemented as a standalone component), or to the Notification Broker/Consumer (in the case where the Notification Cache functionality is embedded in a Broker/Consumer component).

In case of passive synchronization the *Notification Cache* (or the embedding *Notification Broker/Consumer* where the *Notification Cache* functionality is embedded in a *Broker/Consumer* component) MUST use the wsn-bw:Notify operation to feed new *Notification Messages* to the *Notification Consumer*.

No additional rules or limitations are defined other than those specified in the corresponding WSN standard specifications ([OASIS WS-BaseNotification 1.3, 2006], [OASIS WS-BrokeredNotification 1.3, 2006]).

3.3.2.1 Data types

See Section 3.2 of [OASIS WS-BaseNotification 1.3, 2006].

3.3.2.2 Inputs

See Section 3.2 of [OASIS WS-BaseNotification 1.3, 2006].

3.3.2.3 Outputs

See Section 3.2 of [OASIS WS-BaseNotification 1.3, 2006].

3.3.2.4 Errors and faults

See Section 3.2 of [OASIS WS-BaseNotification 1.3, 2006].

3.3.3 Operation CreatePullpoint

This is a request-response operation initiated from the *Notification Requestor* that sends a wsnt:CreatePullPointRequest message to a *Notification Cache*. In response the *Notification Cache* MUST create and activate a new *Pull Point*.

If the *Notification Cache* implements the *Pull-Style Notification*, then it MUST implement the CreatePullPoint operation of the CreatePullPoint interface following the specification given in Section 5.2 of [OASIS WS-BaseNotification 1.3, 2006].



No additional rules or limitations are defined other than the ones specified in [OASIS WS-BaseNotification 1.3, 2006].

3.3.3.1 Data types

See Section 5.2 of [OASIS WS-BaseNotification 1.3, 2006].

3.3.3.2 Inputs

See Section 5.2 of [OASIS WS-BaseNotification 1.3, 2006].

3.3.3.3 Outputs

See Section 5.2 of [OASIS WS-BaseNotification 1.3, 2006].

3.3.3.4 Errors and faults

See Section 5.2 of [OASIS WS-BaseNotification 1.3, 2006].

3.3.4 Operation DestroyPullpoint

This is a request-response operation initiated from the *Notification Requestor* that sends a wsnt:DestroyPullPointRequest message to a *Pull Point*. In response the *Pull Point* MUST attempt to destroy itself and respond with a wsnt:DestroyPullPointResponse message.

If the *Notification Cache* implements the *Pull-Style Notification*, then it MUST implement the DestroyPullpoint operation of the PullPoint interface following the specification given in Section 5.1.4 of [OASIS WS-BaseNotification 1.3, 2006].

Note that the wsnt: DestroyPullPoint message MUST be sent to the uniform resource locator (URL) specified in the wsnt:CreatePullPointResponse/wsnt: PullPoint/wsa:Address element included in the response from the *Pull Point* creation.

No additional rules or limitations are defined other than those specified in [OASIS WS-BaseNotification 1.3, 2006].

3.3.4.1 Data types

See Section 5.1.4 of [OASIS WS-BaseNotification 1.3, 2006].

3.3.4.2 Inputs

See Section 5.1.4 of [OASIS WS-BaseNotification 1.3, 2006].

3.3.4.3 Outputs

See Section 5.1.4 of [OASIS WS-BaseNotification 1.3, 2006].

3.3.4.4 Errors and faults

See Section 5.1.4 of [OASIS WS-BaseNotification 1.3, 2006].

3.3.5 Operation GetMessages

This is a request-response operation initiated from the *Notification Consumer* that sends a wsnt:GetMessagesRequest message to a *Pull Point*. In response, the *Pull Point* MUST attempt to return a wsnt:GetMessagesResponse message containing the *Notification Messages* queued at the *Pull Point*.

If the *Notification Cache* implements the *Pull-Style Notification*, then the created *Pull Point* MUST implement the GetMessages operation of the PullPoint interface following the specification given in Section 5.1.2 of [OASIS WS-BaseNotification 1.3, 2006].



Note, that the wsnt:GetMessagesResponse message. MUST be sent to the URL specified in the wsnt:CreatePullPointResponse/wsnt:PullPoint/wsa:Address element included in the response from the *Pull Point* creation.

No additional rules or limitations are defined other than the ones specified in [OASIS WS-BaseNotification 1.3, 2006].

3.3.5.1 Data types

See Section 5.1.2 of [OASIS WS-BaseNotification 1.3, 2006].

3.3.5.2 Inputs

See Section 5.1.2 of [OASIS WS-BaseNotification 1.3, 2006].

3.3.5.3 Outputs

See Section 5.1.2 of [OASIS WS-BaseNotification 1.3, 2006].

3.3.5.4 Errors and faults

See Section 5.1.2 of [OASIS WS-BaseNotification 1.3, 2006].

3.4 Behaviours Not Specified in the Notification Cache

The goal of this section is to include additional scenarios that are not described in the Notification Cache Technical Specification [NCIA TR/2012/SPW008423/10, 2014].

3.4.1 Pull Points

If the Pull-Style Notification interface is supported, the Notification Cache:

- MUST guarantee the *Pull Point* persistence across application or system restarts.
- MUST guarantee the persistence of the *Notification* in the *Pull Point* queue across application or system restart.
- SHOULD advertise, through policy assertions or other means, the behaviour in case the *Pull Point* queue is no longer able to hold the *Notifications*.

Regarding the way how requestors interact with the *Pull Point* in order to retrieve *Notification Messages* using the GetMessages operation, a *Notification Cache* SHOULD advertise through policy assertions or other means:

- The order in which messages will be delivered by a GetMessages operation.
- The intended behaviour regarding messages that are not delivered because they exceed the value of the MaximumNumber parameter of a GetMessages operation.

There are situations in which a *Pull Point* can be considered unused, for example if the messages accumulated at the *Pull Point* are not retrieved by the *Notification Consumer* for a long time. In such situations the *Notification Cache* can decide to apply specific strategies to terminate the *Pull Point*. In such a situation the *Notification Cache* SHOULD advertise through policy assertions or other means:

- The criteria it uses to decide that a *Pull Point* is unused.
- The actions it takes to proceed with an unused Pull Point.



4 REFERENCES

[IETF RFC 2119, 1997]:

Internet Engineering Task Force (on-line), http://www.ietf.org, Request for Comments 2119, "Key Words for Use in RFCs to Indicate Requirement Levels", S. Bradner, at http://tools.ietf.org/html/rfc2119, March 1997, viewed 31 August 2012.

[NATO TIDE Community, 2012]:

NATO TIDE Community Allied Command Transformation (on-line), http://tide.act.nato.int, "NATO C3 Classification Taxonomy", at http://tide.act.nato.int/tidepedia/index.php?title=

NATO_C3_Classification_Taxonomy, 2012, viewed 31 August 2012. [NC3A RD-3139, 2011]:

NATO Consultation, Command and Control Agency Reference Document 3139, "Publish/Subscribe Service Interface Profile Proposal. Version 1.0" (provisional title), NC3A Core Enterprise Services Team, NC3A, The Hague, Netherlands, unpublished document dated March 2011 (NATO Unclassified).

[NC3A RD-3143, 2011]:

NATO Consultation, Command and Control Agency Reference Document 3143, "Messaging Service Interface Profile Proposal" (provisional title), NC3A Core Enterprise Services Team, NC3A, The Hague, Netherlands, unpublished document dated March 2011 (NATO Unclassified).

[NC3A RD-3292, 2011]:

NATO Consultation, Command and Control Agency Reference Document 3292, "Publish/Subscribe Core Enterprise Service and New User Requirements" (provisional title), D. Gujral, M. Lehmann, V. De Sortis, NC3A, The Hague, Netherlands, unpublished document dated August 2011 (NATO Unclassified).

[NCIA TR/2012/SPW008423/10, 2014]:

NATO Communications and Information Agency Technical Report TR/2012/SPW008423/10, "Notification Cache Technical Specification", V. De Sortis, A. Gruler, M. Lehmann, NCI Agency, The Hague, Netherlands, January 2014 (NATO Unclassified).

[OASIS WS-BaseFaults 1.2, 2006]:

Organization for the Advancement of Structured Information Standards (on-line), http://www.oasis-open.org, "Web Services Base Faults 1.2 (WS-BaseFaults)", L. Liu, S. Meder, at http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf, 1 April 2006, viewed 13 July 2012.

[OASIS WS-BaseNotification 1.3, 2006]:

Organization for the Advancement of Structured Information Standards (on-line), http://www.oasis-open.org, "Web Services Base Notification 1.3 (WS-BaseNotification)", at http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-os.pdf, 1 October 2006, viewed 13 July 2012.

[OASIS WS-BrokeredNotification 1.3, 2006]:

Organization for the Advancement of Structured Information Standards (on-line), http://www.oasis-open.org, "Web Services Brokered Notification 1.3 (WS-BrokeredNotification)", at http://docs.oasis-open.org/wsn/wsn-ws_brokered_notification-1.3-spec-os.pdf, 1 October 2006, viewed 13 July 2012.



[OASIS WSN, 2006]:

Structured Information Standards (on-line), Organization for the Advancement of **"OASIS** TC", https://www.oasis-open.org, Web Notification (WSN) at Services https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsn, 11 October 2006, viewed 13 July 2012.

[W3C WS-Addressing Core, 2006]:

World Wide Web Consortium (on-line), http://www.w3.org, "Web Services Addressing 1.0 – Core", W3C Recommendation, M. Gudgin, M. Hadley, T. Rogers, at http://www.w3.org/TR/2006/REC-ws-addr-core-20060509, 9 May 2006, viewed 13 July 2012.

[W3C XML Namespaces 1.1, 2006]:

World Wide Web Consortium (on-line), http://www.w3.org, "Namespaces in XML 1.1 (Second Edition)", W3C Recommendation, A. Layman, D. Hollander, R. Tobin, T. Bray, at http://www.w3.org/TR/2006/REC-xml-names11-20060816, 16 August 2006, viewed 13 July 2012.

[W3C XPath 3.0, 2011]:

World Wide Web Consortium (on-line), http://www.w3.org, "XML Path Language (XPath) 3.0", W3C Working Draft, D. Chamberlin, J. Snelson, J. Robie, M. Dyck, at http://www.w3.org/TR/2011/WD-xpath-30-20110614/, 14 June 2011, viewed 13 July 2012.



5 ABBREVIATIONS

- Bi-SC Bi-Strategic Command
- BNF Backus-Naur Form
- CES Core Enterprise Services
- NNEC NATO Network Enabled Capability
- SIP Service Interface Profile
- TC Technical Committee
- URL Uniform resource locator
- WSDL Web services description language
- SOAP Simple object access protocol
- XML Extensible markup language



ANNEX 1 - NON-NORMATIVE INTERFACE SPECIFICATION

1.1 Non-Normative WSDL Interface

A NON-normative copy of the web services description language (WSDL) interface definition can be found in the Appendix of the "Notification Cache – Technical Specification" [NCIA TR/2012/SPW008423/10, 2014].

1.2 Non-Normative XML Schema Definition

A NON-normative copy of the extensible markup language (XML) Schema definition can be found in the Appendix of the "Notification Cache – Technical Specification" [NCIA TR/2012/SPW008423/10, 2014].