2020-03

ARCHITECTING AUTONOMOUS ACTIONS IN NAVY ENTERPRISE NETWORKS

Geiszler, Max M.

Monterey, CA; Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

ARCHITECTING AUTONOMOUS ACTIONS IN NAVY ENTERPRISE NETWORKS

by

Max M. Geiszler

March 2020

Thesis Advisor:	Dan C. Boger
Co-Advisor:	Luqi
Second Reader:	Sharon M. Runde

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2020	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
ARCHITECTING AUTONOMOUS ACTIONS IN NAVY ENTERPRISE NETWORKS

5. FUNDING NUMBERS

6. AUTHOR(S) Max M. Geiszler

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Fleet Cyber / 10th Fleet, Fort Meade, MD

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT	**12b. DISTRIBUTION CODE**
Approved for public release. Distribution is unlimited.	A

13. ABSTRACT (maximum 200 words)

The three Navy Enterprise Networks (NEN) IT-21, NMCI, and ONE-NET have a slew of automation integration work required in order to match their modern commercial equivalents in the mission of Network Operations (NetOps). Commercial companies such as AT&T, Amazon, Verizon, Extreme Networks, etc., have adopted network automation in their practice of NetOps, which have reduced manpower and increased network vigilance. This thesis shows how the Navy currently utilizes "reactive" network process controls in the conduct of NetOps by describing multiple-use–cases of reactive processes currently practiced by NENs. It then shows that there are two fundamental changes to NENs necessary in order to transition NENs into a state of "pro-active" service, by qualitative analysis of good practices by industry and smaller Navy organizations. With a goal to ensure that NENs are able to anticipate and react before predictable problems arise, the two changes suggested by this thesis are to consolidate Navy operational data and employ an "automation framework" to enable Development Operations (DevOps) practices in the conduct of NetOps. Last, a short- and long-term solution for changes to NEN architectures to facilitate these two changes are then presented.

14. SUBJECT TERMS	**15. NUMBER OF PAGES**
automation, Navy, enterprise, network, integration, infrastructure, re-active, INOSS, development, operations, DEVOPS	137
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

ARCHITECTING AUTONOMOUS ACTIONS IN NAVY ENTERPRISE NETWORKS

Max M. Geiszler
Lieutenant, United States Navy
BS, Oregon State University, 2010

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2020

Approved by: Dan C. Boger
Advisor

Luqi
Co-Advisor

Sharon M. Runde
Second Reader

Peter J. Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The three Navy Enterprise Networks (NEN) IT-21, NMCI, and ONE-NET have a slew of automation integration work required in order to match their modern commercial equivalents in the mission of Network Operations (NetOps). Commercial companies such as AT&T, Amazon, Verizon, Extreme Networks, etc., have adopted network automation in their practice of NetOps, which have reduced manpower and increased network vigilance. This thesis shows how the Navy currently utilizes "reactive" network process controls in the conduct of NetOps by describing multiple-use–cases of reactive processes currently practiced by NENs. It then shows that there are two fundamental changes to NENs necessary in order to transition NENs into a state of "pro-active" service, by qualitative analysis of good practices by industry and smaller Navy organizations. With a goal to ensure that NENs are able to anticipate and react before predictable problems arise, the two changes suggested by this thesis are to consolidate Navy operational data and employ an "automation framework" to enable Development Operations (DevOps) practices in the conduct of NetOps. Last, a short- and long-term solution for changes to NEN architectures to facilitate these two changes are then presented.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

Glossary

Agile Core Services

ACS is a combination of Commercial Off the Shelf (COTS) products integrated by Space and Naval Warfare Systems Center Pacific (SSC Pacific) under the Tactical Networks Program Office (PMW 160). ACS is a sub-system of the Consolidated Afloat Networks and Enterprise Services (CANES) program and is defined by the need to provision loosely-coupled systems and capabilities to the warfighter, which can be reused, discovered, and linked to build mission threads. ACS provides the warfighter with common enterprise solutions to improve information sharing and combat effectiveness. The suite of capabilities facilitates information access and exchange to allow dynamic collaboration and accelerated decision-making amid unprecedented quantities of operational data. ACS 3 provides a set of common software infrastructure services for common core services, data ingest, data representation, data storage, data indexing, data access, and framework to support analytics. ACS directly enables Department of Defense, Navy, Program Executive Office for Command, Control, Communications, Computers and Intelligence (PEO-C4I), and SPAWAR strategies for rapid delivery, application architecture, data strategy, cloud, and development to operations (DevOps) software engineering best practices. [GLS1]

Authorization to Operate

The official management decision given by a senior organizational official to authorize operation of an information system and to explicitly accept the risk to organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, and the Nation based on the implementation of an agreed-upon set of security controls and privacy controls. [GLS2]

Commander 10th Fleet

U.S. TENTH Fleet is the operational arm of Fleet Cyber Command and executes its mission through a task force structure similar to other warfare commanders. In this role, TENTH Fleet provides operational direction through the command's Maritime Operations Center located at Fort George Meade, Md. [GLS3]

Consolidated Afloat Networks and Enterprise Services

CANES is the consolidation and enhancement of the requirements for five existing legacy network programs, as well as a single support framework for all C4I applications that currently require dedicated infrastructure to operate delivered and managed legacy systems. These include the Integrated Shipboard Network System (ISNS), Sensitive Compartmented Information (SCI) Networks, and Combined Enterprise

Regional Information Exchange System Maritime (CENTRIXS-M). The CANES concept requires a technical and programmatic realignment of afloat infrastructure and services. CANES will take advantage of the new business model of open architecture, Service Oriented Architecture (SOA), and rapid COTS insertion, in order to bring fiscal savings to the Navy, as well as operational agility to the warfighter. [GLS4]

Defense Intelligence Information Enterprise DevTools

DI2E DevTools has become the first system at NRO to receive an authority to operate (ATO) in the unclassified, AWS GovCloud environment. These tools are provided for Government programs and their supporting developers. The DI2E Developer Collaboration Tools provide documentation and design artifact hosting, issue tracking, and project collaboration. The tools enable automating nightly builds and tests of software projects flagging and sending errors to development team(s) for action, automating unit and IVV tests sending results to development team(s) upon completion, check-in/check-out capability to maintain versioning of software baselines in shareable source code and design artifact repositories. The DI2E Developer Collaboration Tools provide an environment where code produced for the government is configuration managed and buildable on government owned property. [GLS5]

Fleet Cyber Command

U.S. Fleet Cyber Command reports directly to the Chief of Naval Operations as an Echelon II command and is responsible for Navy information network operations, offensive and defensive cyberspace operations, space operations and signals intelligence. As such, U.S. Fleet Cyber Command serves as the Navy component command to U.S. Cyber Command, the Navy space component to U.S. Strategic Command, and the Navy's Service Cryptologic Component Commander under the National Security Agency/Central Security Service. U.S. TENTH Fleet is the operational arm of Fleet Cyber Command and executes its mission through a task force structure similar to other warfare commanders. In this role, TENTH Fleet provides operational direction through the command's Maritime Operations Center located at Fort George Meade, Md. [GLS3]

Higher Headquarters

A military unit's superior in the operational chain of command.

Information Technology for the 21st Century network

IT-21 is an information transfer strategy that provides network connectivity capable of voice, data and video for afloat units.

Navy Enterprise Network

NEN is the PMW 205 program office, but also a common term used to describe the

three major networks for the Navy. Those networks are called IT-21, NMCI, and Outside Continental U.S. (OCONUS) Navy Enterprise Network (ONE-NET).

Network Operations

The DoD-wide operational, organizational, and technical capabilities for operating and defending the Global Information Grid (GIG). NetOps includes, but is not limited to, enterprise management, net assurance, and content management. NetOps provides commanders with GIG situational awareness to make informed command and control decisions. GIG situational awareness is gained through the operational and technical integration of enterprise management and defense actions and activities across all levels of command (strategic, operational, and tactical.) [GLS6]

Network Operations Center

Responsible for operating, maintaining, and securing the Navy's CONUS Navy/Marine Corps Intranet network, the same type of function can be referred to as a Fleet Network Operations Center (FLTNOC) for afloat Information Technology for the 21st Century network or Theater Network Operations and Security Center (TNOSC) for ONE-NET.

OCONUS Navy Enterprise Network

Evolved from the Base Level Infrastructure Information (BLII) Modernization Program in 2005, ONE-Net provides secure, seamless and global computer connectivity for the DON outside the continental US. [GLS7]

Casualty Report

A Navy message that reports system casualties that degrade/fails a Navy Units mission and or warfare area that they are responsible for.

Fleet Broadcast

A system employed by the Navy to send message traffic.

NOVA

A Navy message traffic management tool which will read in message traffic, check its format, and send it to correct destination if no errors detected, or saves message to be dealt with by manual human NOVA operator when the system has an error.

Operator

A member of the NetOps team which understands a network's mission context and is employed in such a way that actions taken to accomplish any objectives are in effect accomplishing the mission of the organization for which they are employed. An operator does not always have a strong understanding of the underlining system's architecture.

THIS PAGE INTENTIONALLY LEFT BLANK

Glossary References

[GLS1] K. Rohler. (2019, Aug). Agile Core Services (ACS). Naval Information Warfare Center Pacific. [Online]. Available: https://confluence.di2e.net/display/ACS3EXT/Agile+Core+Services+3+External+Support+Portal

[GLS2] Authorization to Operate. (n.d.). COMPUTER SECURITY RESOURCE CENTER(CSRC). [Online]. Available: https://csrc.nist.gov/glossary/term/authorization-to-operate. Accessed Oct. 31, 2019.

[GLS3] WELCOME TO FCC/C10F. (n.d.). U.S. Fleet Cyber Command. [Online]. Available: https://www.public.navy.mil/fcc-c10f/Pages/home.aspx. Accessed Oct. 31 2019.

[GLS4] B. Gallo. (n.d.). Consolidated Afloat Networks and Enterprise Services (CANES). [Online]. Available: https://www.public.navy.mil/navwar/technology/Pages/ConsolidatedAfloatNetworksandEnterpriseServicesCANES.aspx

[GLS5] Frequently Asked Questions. (n.d.). Defense Intelligence Information Enterprise. [Online]. Available: https://www.di2e.net/display/DI2E/DI2E+DevTools. Accessed Dec. 05, 2019.

[GLS6] J. Grimes, *NetOps for the Global Information Grid (GIG)*, DODI 8410.02, DOD CIO, Washington, DC, 2012. Available: https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/841002p.pdf

[GLS7] Program Executive Office for Enterprise Information Systems (PEO EIS). (n.d.). Naval Information Warfare Systems Command. [Online]. Available: https://www.public.navy.mil/navwar/PEOEIS/Pages/_NEN.aspx. Accessed May. 16 2019.

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

ACL	Access Control List
ACS	Agile Core Services
ADCON	Administrative Chain of Command
AI	Artificial Intelligence
API	application programming interface
ATO	Authorization to Operate
AWS	Amazon Web Services
BWC	Battle Watch Captain
C10F	Commander 10th Fleet
C2	Command & Control
C2C24	Compile to Combat in 24 Hours
C2OIX	C2 Official Information Exchange
CANES	Consolidated Afloat Networks and Enterprise Service
CASREP	Casualty Report
CCB	Change Control Board
CCIR	Commander's Critical Informational Requirement
CMS	Call Management System

CNO	Chief of Naval Operations
COC	Chain of Command
COMSPOT	Communications Spot Report
CONOPS	Concept of Operations
CONUS	Inside Continental U.S.
COP	Common Operational Picture
COS	Chief of Staff
COTS	commercial off the shelf
CUDIXS	Common User Digital Information Exchange Subsystem
DCO	Defensive Cyberspace Operations
DevOps	Development Operations
DEVSECOPS	Development Security Operations
DI2E-DEVTOOLS	Defense Intelligence Information Enterprise
DISA	Defense Information Systems Agency
DoD	Department of Defense
DODIN-N	Department of Defense Information Network - Navy
ECMD	Enterprise Content Management & Delivery
ENMS	Enterprise Network Management System

FACE	Future Airborne Capability Environment
FAO	Fleet Authorization Official
FCC	Fleet Cyber Command
FLTNOC	Fleet Network Operations Center
FMX	Fleet Message Exchange
FOUO	For Official Use Only
GEP	Global Enterprise Partners
GIG	Global Information Grid
GNOC	Global Network Operations Center
GSSC	Global Satellite Communications (SATCOM) Support Center
GUI	Graphical User Interface
HCI	human computer interactions
HHQ	Higher Headquarters
HP	Hewlett Packard
HPE	Hewlett Packard Enterprise
HPSM	Hewlett Packard Service Manager
IAC	Infrastructure as Code
ICMP	Internet Control Message Protocol

INOSS	Integrated Navy Operations Support System
IPS	Intrusion Prevention System
ISPi	Hewlett Packard (HP) Network Node Manager i Smart Plug-in
IT	Information Technology
IT-21	Information Technology for the 21st Century network
ITSM	Information Technology (IT) Service Management
JFHQ-DODIN	Joint Force Head Quarters - Department of Defense (DoD) Information Network
JFTOC	Joint Fleet Telecommunications Operations Center
MUA	Master Update Authority
MUOS	Mobile User Objective System
MVP	minimum viable product
NAO	Navy's Authorization Official
NAVAIR	Naval Air Systems Command
NAVIFOR	U.S. Naval Information Forces
NAVSEA	Naval Sea Systems Command
NCDOC	Navy Cyber Defense Operations Command
NCSA	Navy Cyber Situational Awareness

NCTAMS	Naval Computer and Telecommunications Area Master Station
NCTAMS-LANT	NCTAMS - Atlantic
NCTAMS-PAC	NCTAMS - Pacific
NCTS	Naval Computer and Telecommunications Station
NEN	Navy Enterprise Network
NetOps	Network Operations
NGEN-R	Next Generation Enterprise Networks Re-compete
NIWC	Naval Information Warfare Command
NMCI	CONUS Navy/Marine Corps Intranet
NNMi	Network Node Manager version i
NNWC	Navy Network Warfare Command
NOC	Network Operation Center
NOVA	the automated message store and forward system (NOVA)
NPS	Naval Postgraduate School
NTP-4	Naval Telecommunications Procedures - 4
OCONUS	Outside Continental U.S.
ONAP	Open Network Automation Platform

ONE-NET	OCONUS Navy Enterprise Network
OPNAV	Office of the Chief of Naval Operations
OSI	Open Systems Interconnection
OSS	Operations Support System
PEO	Program Executive Office
PLA	Plain Language Address
POR	Program of Record
QA	Quality Assurance
RMF	Risk Management Framework
ROC	request operations center
RSSC	Regional SATCOM Support Center
RYBY	Rules You Build Yourself
SA	Situational Awareness
SAC	Security as Code
SATCOM	Satellite Communications
SDLC	software development life cycle
SDN	Software Defined Network
SI	Software Engineering Institute

SLA	Service Level Agreement
SLR	Service Level Requirement
SME	Subject Matter Expert
SNDL	Standard Navy Distribution List
SOC	Security Operations Center
SSIXS	Submarine Satellite Information Exchange Subsystem
SYSCOM	Systems Command
TECHREP	Technical Representative
TMS	Trouble Management System
TNOSC	Theater Network Operations and Security Center
TRB	Technical Review Board
VM	Virtual Machine
VNF	Virtual Network Function
VRAM	Vulnerability Remediation Asset Manager
WO	Watch Officer

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

This thesis hypothesizes that Navy Enterprise Networks (NENs) have reactive process management to everyday Network Operations (NetOps) actions. Reactive process management in NetOps means network operators, who make up a 24/7 watch at Network Operations Centers (NOCs), must wait until a problem occurs before they are able to implement mitigations..

It further suggests that the two reasons responsible for this lack of proactiveness are quantitatively shown to be due to:

1. lack of capability for all Programs of Record (PORs) to access information repositories via a shared declarative application programming interface (API) solution that is developed and maintained by a single Navy organization

2. lack of an automation framework to enable NENs to do Development Operations (DevOps), provide fast deployment test capability for application update integration, and focus the culture of Navy enterprise network operations to adopt DevOps as a fundamental practice

The multiple use-case examples in 4 support Navy NetOps' reliance on reactive process management. Chapter Use-cases were taken from observation after multiple site visits to Navy NOCs. The first case covered shows a lack of proactive processes in Perspecta's request operations center (ROC) where multiple analysts are required to synchronize information after problematic tickets arrive, without a means to streamline or correlate repetitive ticketing situations. The second case covers Inside Continental U.S. (CONUS) Navy/Marine Corps Intranet (NMCI) and how its NetOps reporting procedures cannot be streamlined due to lack of an automation framework and a confusing system of multiple locations where operational ticketing information is stored. The third case is over the afloat Information Technology for the 21st Century network (IT-21) message traffic system, which has essentially no automative capabilities and has an outdated slow reliance on Communications Spot Report (COMSPOT) message traffic to relay rudimentary information. The fourth case covers the neglect in using Mobile User Objective System (MUOS) satellite logs to proactively seek

and solve problems before they occur. The fifth case covers POR architectural problems and shows how PMW-130 is overextended in trying to provide the Navy with the capability of warehousing data after hunting it down and building unique capabilities to discover and retain it.

The two reasons responsible for this lack of proactive processes are then qualitatively shown in Chapter 5. The commercial practices discussed are the evolution of how DevOps was discovered by AT&T and the tool suite offered by Extreme Networking are used to enhance the DevOps capability of organizations.

The smaller Navy organizations found to further show the qualitative conclusion of the thesis are with the Naval Postgraduate School (NPS), an effort from Perspecta, and an effort from Navy Network Warfare Command (NNWC). NPS has built its is shown to have built its own solution to an automation framework and used Splunk to consolidate and facilitate a shared API solution for data pooling. With Perspecta it is shown to have created a venture to compete for the Next Generation Enterprise Networks Re-compete (NGEN-R) contract in the implementation of DevOps within Perspecta as a proof of necessity. Lastly an effort by NNWC to provide the Navy with a data-lake capability is covered as further evidence that the Navy needs a universally consolidated data solution with an API solution in place.

The thesis then recommends a short- and long-term solution to these issues in Chapter 6.

The short-term recommendation covers the API issue through full adoption of Splunk, and may possibly solve the automation framework with implementation of DevOps by adopting a maturing Consolidated Afloat Networks and Enterprise Service (CANES)/Agile Core Services (ACS) solution.

The long-term recommendation suggests the adoption and implementation of NNWC's Integrated Navy Operations Support System (INOSS) framework with some enhancements. Enhancements include a change to the framework to fully adopt automation and the adoption of the integration framework known as Future Airborne Capability Environment (FACE), which would allow the INOSS framework to adopt any FACE compliant third-party software to include software developed by joint government organizations that have already been integrated with FACE.

Acknowledgments

This thesis could not have been written without the funding of Fleet Cyber Command (FCC) to further the knowledge of Navy network health and status, which enabled site visitations, and the funding of Amazon Web Services (AWS) lab research and material.

To Dr. Luqi, thank you for your persistent drum beat in getting more material written, allowing me to use the Software Engineering lab, and your desire for constant updates. You were the only reason the AWS development of the Consolidated Afloat Networks and Enterprise Service (CANES)/Agile Core Services (ACS) prototype was possible, and pretty much the only reason I finished this thing on time.

To Dr. Boger, thank you for being steadfast in your availability, giving solid advice at every corner, and supporting our team of researchers even when we veered off track ... many times.

To Sharon Runde, thank you for your help in facilitating the meetings, proofreading everything, and assisting in getting purchases. You made life easier and that is always appreciated.

To Robert Sweeney, thank you for taking time away from your primary job to teach multiple classes and mentor me throughout this entire project. If it was not for your many hours of instruction on software architecture, Chapter 6 would not have been possible.

To Bruce Chiarelli, thank you for providing expert knowledge in building out the CANES architecture in AWS. Your effort and drive is the underlying reason the cluster will continue to exist into the the future at NPS, which, in turn, will allow for focused research in one of the few Navy ship application prototypes that can be studied in an academic environment.

To Jasmine Mally from the Graduate Writing Center (GWC), thank you "SO MUCH!" (with the terrible emphasis I used and you kept getting rid of). If it was not for your ten (not "10" because numbers start at 11 in IEEE format) hours of revising this thesis, probably no one would understand any word of it!

And to my wife, Beverly Geiszler, none of this was feasible without your support at home.

Thank you so much for allowing me to work late many nights, and being my editor-in-chief when I had to plug in all of the grammatical corrections from the GWC and Dr. Luqi. You saved my eyes and brain! I love you.

CHAPTER 1:
Navy Enterprise Network Automation

1.1 Problem Description

While commercial industries for the last ten years have invented newer methods to operate networks through automated processes, workflow orchestration, and rapid network healing capabilities, Navy Enterprise Networks (NENs) today still rely heavily on manual processes. The future of Navy networks depends on reducing repetitive human in the loop actions.

The network which Navy entities operate must devote numerous man-hours in order for it to be run at a mediocre capacity, and resources and effectiveness of actions employed by NENs are poorly managed. Most Navy operations currently depend on a networked solution, and many network operator man-hours are spent accomplishing necessary mundane tasks. Actions in NEN environments are performed without a systematic approach to automate the actions necessary for similar situations in the future.

Autonomous technological procedures are essential in the conduct of Network Operations (NetOps), and they are the only way to provide speed and pro-activeness in a networked environment. The goal of this thesis is to provide suggestions for a foundational architecture of NENs, which would enable the crafting of autonomous actions at the Network Operations Center (NOC) level, pull humans out of repetitive task processes, and overall increase effectiveness of NENs by creating a pro-active environment to operate in.

This study exposes the multiple linkages and separations between the way large commercial enterprise networks operate and scale network automation, and the way the Navy currently utilizes network automation. To establish some of these background linkages, this thesis will cover a large breadth of basic Navy internal workings/processes and the basic concept of Development Operations (DevOps).

1.1.1 Hypothesis and Proposed Changes

This thesis targets the structure of Navy Network automation through the practice of Navy NetOps. Specifically, the goal is to first show why the current NetOps functions at NENs are not able to function at a speed fast enough to get the Navy to a state of "proactiveness." This is shown through use of current use-case examples, and an explanation of why the Navy cannot anticipate problems in a network without first adopting two key changes that are already employed by commercial industries conducting NetOps.

Two proposed changes after a quantitative analysis of industry practices, and smaller thriving accepted Navy networks are shown to solve this problem. The first proposed change is to enable Navy networks to transition into proactive network operations to standardize the Navy's network data storage architecture and how it combines and pools information. This includes providing a capability for all Programs of Record (PORs) to access information repositories via a shared declarative application programming interface (API) solution that is developed and maintained by a single Navy organization.

The second change targets the Navy's software development business process. Building into all software development lifecycles, a DevOps environment that can quickly implement necessary technical changes in all NEN architectures to facilitate faster automation. Specifically, the focus is on how DevOps allows for faster operational information flow to Higher Headquarters (HHQ). DevOps may also drive some of the underlying architectural changes to Navy networks, as it inherently can find the network limitations to the environment in which it is employed. What is specifically needed in this change is the re-tooling of networks to enable DevOps, a fast deployment test capability for doing application update integration, and lastly focusing on the culture of Navy enterprise network operations to adopt DevOps as a fundamental practice.

1.2 Narrowing the Focus

This thesis addresses the structure of Navy network automation in the context of the practice of Navy NetOps. Specifically, the goal is to first show why the current NetOps functions at NENs will not be able to function at a speed fast enough to enable Navy networks to transition to proactive network operations. This is shown through use-case examples, and an explanation of why the Navy will not be able to anticipate problems in a network

without first adopting two key changes that are already employed by commercial industries conducting NetOps.

The first of the proposed changes is to the Navy's network data storage architecture and how it combines and pools its information. This includes providing the capability for all PORs to access information repositories via a shared declarative API solution developed and maintained by a single Navy organization.

The second change targets the Navy's software development business process. If built into all software development lifecycles, a DevOps environment that can quickly implement necessary technical changes in all NEN architectures to facilitate faster automation. Specifically, the focus will be on how DevOps allows for faster operational information flow to HHQ. DevOps may also drive some of the underlying architectural changes to Navy networks, as it inherently can find the network limitations of the environment in which it is employed. Specifically, this change needs:

1. The re-tooling of networks to enable DevOps
2. A fast deployment test capability for doing application update integration
3. Focusing on the culture of NEN operations to adopt DevOps as a fundamental practice.

This thesis shows these steps to be necessary to bring the modern NEN into an age of proactive NetOps employment. We acknowledge that policies and potentially Navy acquisition processes may also need to be changed in order to adopt the two proposed changes. This thesis does not address that aspect of this problem, but provides many recommendations for future areas of research.

The content in this thesis is based on the information collected from site visits and interviews between dates of June 2018–December 2019 and may be subject to change by publication date March of 2020. NEN environments are constantly in a state of flux, and multiple new technological implementations were "in the works" at the time of each site visit, so it is possible for some of the current information pertained in this thesis to be outdated by time of publication.

1.2.1 Thesis Outline

Chapter 2

This NEN automation research focuses solely on automation of actions at the NOC level, and it is a piece of a larger research project conducted by Naval Postgraduate School (NPS) in establishing how to gather "Network Health and Status" information to assist in the mission of NetOps. Chapter 2 outlines the project this thesis fits in its entirety.

Chapter 3

Since the use of operational information in a network and employment of automated procedures are vital to the solution presented in this thesis, the reader needs to understand the Navy's current processes employed for NetOps. Necessary topics include a basic understanding of how the Navy is architected to conduct NetOps, and the major organizational players in this mission. A definition of Navy NetOps is given, and the concept of DevOps is outlined in Chapter 3.

Chapter 4

The Navy operates the largest intranet in the world. Many times, problems are pocketed into multiple network use-cases which do not necessarily affect the Network as a whole, but nonetheless all are vitally important to the mission of Navy entities. In order to describe isolated problems which affect NetOps, multiple use-cases are explained in Chapter 4 to outline some of the issues that currently do not have automated solutions and discuss impacts to the mission of NetOps.

Chapter 5

From commercial practice and small pockets of developing Navy infrastructure, Chapter 5 shows that the Navy cannot conduct NetOps proactively until there is a solution put in place to allow for the practice of DevOps, which thrives best when enabled by the network architecture. The end of this chapter gives a qualitative analysis to validate the hypothesis.

Chapter 6

There is no master plan to organize NEN architecture. Due to the patchwork development of capabilities by pocketed commands and poorly regulated Navy PORs, the Navy has

developed/acquired multiple capabilities but has no ability to employ them effectively across its vast cyberspace expanse. Chapter 6 describes a potential solution to fix a small piece of a larger issue through the use of a consolidated architecture called Integrated Navy Operations Support System (INOSS), and the placement of an "automation framework" in support of DevOps procedures at key network locations.

Chapter 7

This chapter summarizes the findings of this thesis and multiple avenues for future research.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Fitting into a Larger Project

2.1 Navy's Cyber Vision[1]

"Our Navy will protect America from attack, promote American prosperity, and preserve America's strategic influence. U.S. naval operations—from the seafloor to space, from the blue water to the littorals, and in the information domain—will deter aggression and enable resolution of crises on terms acceptable to the United States and our allies and partners" [2]. This is part of the mission statement that was promulgated by Admiral John M. Richardson, prior Chief of Naval Operations (CNO) in *A Design For Maintaining Maritime Superiority*. From this statement one can gather that now, more than ever, the information domain plays a vital role in protecting our country. Within the information domain falls the cyber domain and Fleet Cyber Command (FCC)/Commander 10th Fleet (C10F), which have been charged with defending and delivering effects in and through cyberspace. From FCC's command page, part of their mission is to "conduct operations in and through cyberspace, the electromagnetic spectrum, and space to ensure Navy and Joint/Coalition freedom of action and decision superiority while denying the same to our adversaries" [3]. FCC and C10F are co-located in Fort Meade, MD and also have a dual-hatted commander.

2.2 FCC/C10F Tasking[1]

In 2018, then FCC/C10F Chief of Staff (COS) Captain James Mills directed the N9 to head up research on visualization of data, human computer interactions (HCI), and identify the most efficient way to display network health and status to a watch stander and decision maker. Specific guidance included:

1. Using the INOSS architectural framework, evaluate existing software tools for deployment to Department of Defense Information Network - Navy (DODIN-N) watch floors, NOCs, and for use by afloat Information Technology (IT) personnel.

[1]These sections are a collaborative effort among multiple thesis students working on the same C10F funded project. The thoughts, ideas, and information contained within are not attributed to a single individual and can be found in the theses referenced in the subsections under Section 2.2.

2. Consider industry and other government implementations, best practices, and employment of similar systems.

3. Integrate existing and novel malicious activity notifications into the INOSS framework to allow appropriate personnel the freedom to quarantine and investigate the activity.

4. Identify how these tools will support existing Navy programs of record.

This led to the funding for NPS research to be conducted by thesis students. Dr. Dan Boger was appointed as the project manager and then selected students with skill sets that aligned with the requirements of the aforementioned research. The theses and their respective deliverables are as follows:

2.2.1 Architecting Autonomous Actions in Navy Enterprise Networks

Dr. Dan Boger and Dr. Luqi are co-thesis advisors for Lieutenant Max Geiszler. This thesis investigates NENs in an attempt to better understand the fundamental operation of the Navy's networks. The main idea behind the research is to explain how NENs can conduct NetOps to meet unique Navy mission sets and ensure adequate information is given to higher up organizations. The investigation covers some of the use-cases in which the Navy has intensive need for human driven processes to accomplish necessary critical tasks. It also explores where man-hours are being inefficiently spent due to process redundancy and limited human watch-stander proficiency. It then suggests a technical architectural change to NEN infrastructure utilizing the INOSS framework, which helps to facilitate automated solutions to problems that have been presented by FCC/C10F. It also suggests a change to tightly integrate DevOps in operational processes.

2.2.2 Network Traffic Anomaly Detection on a Navy Network

Dr. John Monaco is the thesis advisor for Lieutenant Mike Laws and Lieutenant Greg Bunder. This thesis determines the viability of using existing unsupervised machine learning techniques to detect anomalous network traffic from an Unclassified Navy network. Upon completion this thesis gives a recommendation as to whether unsupervised machine learning can be used for anomaly detection. If the hypothesis is accurate, detailed analysis of implemented features that are most effective for anomaly detection, along with any lessons

learned and obstacles met during research, are provided. Lastly, this thesis addresses what an architecture might look like that would be used to implement network anomaly detection via unsupervised within the INOSS framework.

2.2.3 Visualization: Functional and Conceptual Approach to Network Operations

Dr. Dan Boger is the thesis advisor for Commander Henry Lee Bush. The thesis analyzes the current visualization at various organizations, the private sector and the public sector, to better understand how visualization provides network's heath and status. The main idea behind the thesis is to evaluate visualization in key focus areas: single pane of glass, information immersion, information framework and information concept. It does this by covering case studies that were done through the site observation to identify how information is collected, processed, analyzed and visualized to support command and control of the network. Through the case studies, the thesis also reviews the information not captured because of stovepiped systems, limited shared management information and manual process which reduces the information in visualization. The information not captured in turn impacts situation awareness and decision making which negatively impacts command and control of the network. The thesis recommends the use of the INOSS functional framework to improve processes to support visualization of information and information immersion through space design. Lastly it introduces an information management concept to support command and control of the network.

2.2.4 How Information Sharing Affects Network Operations

Dr. Dan Boger is the thesis advisor for Lieutenant Eva Castillo. Effective information sharing between various components are crucial to FCC/C10F successfully and efficiently meeting the mission. The thesis has two goals. The primary goal is to examine whether existing information systems, mandates, policies, or Service Level Agreements (SLAs) are limiting information sharing within the FCC/C10F organization. The secondary goal is to seek technical and non-technical solutions to support current and evolving requirements. The thesis will evaluate solutions studied that can positively impact information sharing for the organization. Research approaches include: interviews and observations in academia, civilian IT sector, defense organizations, programs of record, and Tier 1, 2 and

3 providers. From the research gathered, conclusions were drawn to the effectiveness of current technologies, mandates, and policies along with proposed solutions.

CHAPTER 3:
Navy Network Operations

This section covers the basic flow of how operational information is supposed to traverse Navy Enterprise Networks (NENs). It then introduces Development Operations (DevOps) as one of the key components missing in all current NEN architectures.

3.1 Navy's Concept of Network Operations

Network Operations (NetOps) is, first and foremost, the management of Navy networks through all Navy Network Operations Center (NOC) entities. Navy Network Warfare Command (NNWC) and Navy Cyber Defense Operations Command (NCDOC) must then correlate the operational information and continue to pass correlated information to Fleet Cyber Command (FCC)/Commander 10th Fleet (C10F) then up to Joint Force Head Quarters - Department of Defense (DoD) Information Network (JFHQ-DODIN) to be disseminated to the rest of the DoD for optimal operational awareness of DoD networks. Figure 3.1 illustrates this process.

The Navy uses three NENs: the afloat Information Technology for the 21st Century network (IT-21), the Inside Continental U.S. (CONUS) Navy/Marine Corps Intranet (NMCI), and the Outside Continental U.S. (OCONUS) Navy Enterprise Network (ONE-NET). All other networks are considered "excepted networks." For example, Naval Postgraduate School (NPS)'s network is considered one of the excepted networks. This thesis specifically focuses on use-cases for automation in the IT-21 and the NMCI networks.

Every NEN consists of several commands spread out geographically that are generically referred to as a NOC. However, each NEN has a different name to which they refer for their own commands that perform the same functions when it comes to NetOps. The IT-21 calls its commands Fleet Network Operations Centers (FLTNOCs), ONE-NET calls its commands Theater Network Operations and Security Centers (TNOSCs) and NMCI refers to its commands as simply NOCs. Even though each NEN has a different name for its multiple NOCs, all NENs have the same mission in which they conduct NetOps. Today, all Navy NOCs utilize a patchwork of decentralized methods for NetOps, which consists

primarily of phone calls and email reporting of information up to NNWC and NCDOC.

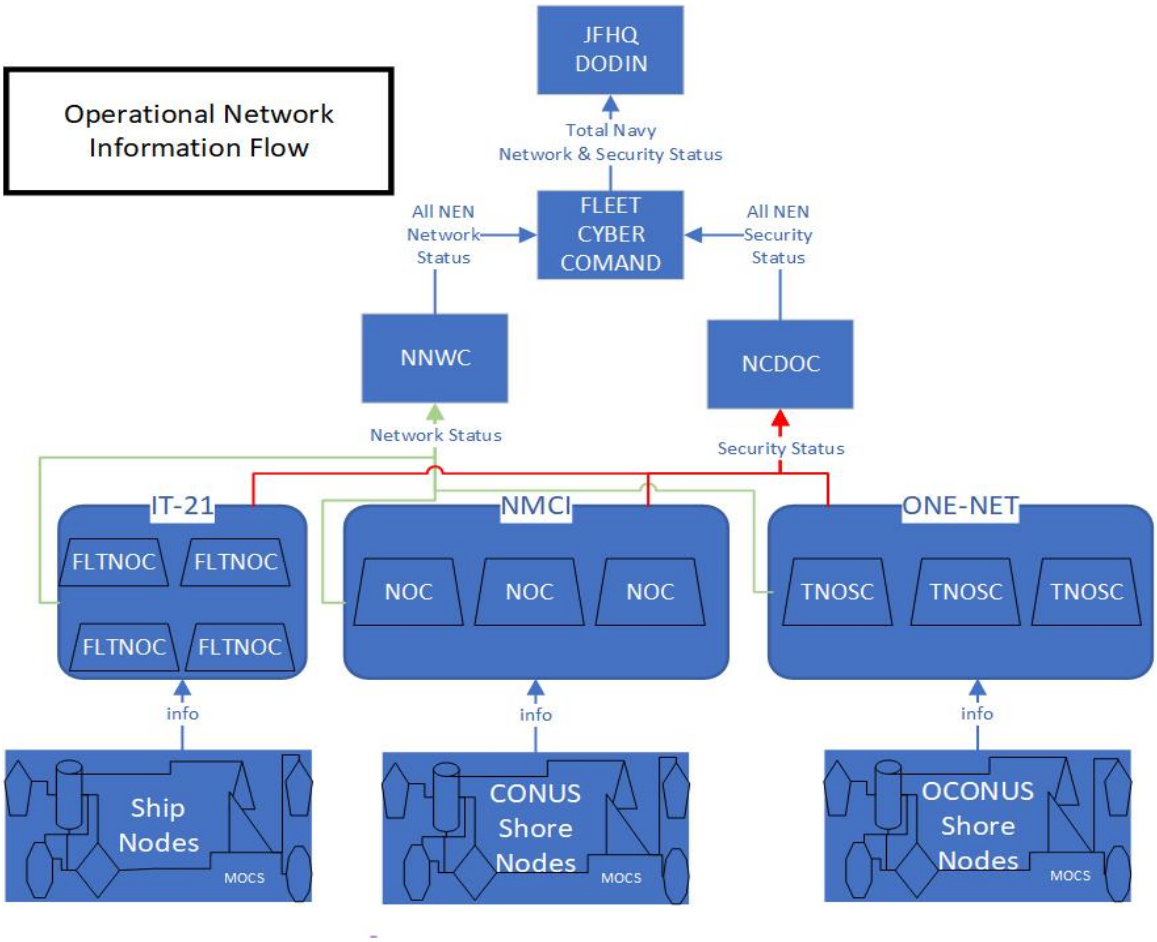


Figure 3.1. How Operational Information Moves Up to Higher Headquarters

Note in Figure 3.1 how all NENs split the types of information funneled into the two organizations as "Network Status" data and "Security Status" data. This is primarily because Network operations are complicated by the fact that operating the network securely is necessary in order to accomplish most any mission set, and there is enough work in that field to devote an entire organization to the task.

NNWC's mission is described as to "operate and defend the Navy's portion of the Global Information Grid and to deliver reliable, secure Net-centric and Space war fighting capabilities in support of strategic, operational and tactical missions across the Navy" [4]. The other organization is NCDOC whose mission statement is to "execute defensive cyberspace

operations, and enable global power projection through proactive network defense" [5].

3.1.1 Navy's Effectiveness of NetOps Today

The DoD Instruction 8410.02 defines NetOps as:

> The DoD-wide operational, organizational, and technical capabilities for operating and defending the Global Information Grid (GIG). NetOps includes, but is not limited to, enterprise management, net assurance, and content management. NetOps provides commanders with GIG situational awareness to make informed command and control decisions. GIG situational awareness is gained through the operational and technical integration of enterprise management and defense actions and activities across all levels of command (strategic, operational, and tactical). [6]

In other words, NetOps is a term used to describe being vigilant and taking control of network related issues that affect any Navy organization whose mission is dependent on the network (aka the GIG). NetOps for the Navy is currently very human intensive work. When network issues arise, most are solved by email or getting on the phone once they have been discovered by the responsible network operator. Operator is a term used to describe a member of a NetOps team. The discovery of issues is another terribly slow human intensive process, which is primarily attributed to the lack of automation of getting information packaged and stored into databases and automation of accessibility, correlation, and recommendations, which we can refer to as "processing of information" inside NENs.

3.1.2 Scope of Information Technology (IT) Operations

The responsibility for NetOps ultimately falls on the shoulders of the IT staff. According to *DevOps A Software Architect's Perspective* a Carnegie Mellon Software Engineering Institute (SI) series book, the following list of duties must be covered by a traditional Operations staff:

- Analyzing system logs and identifying potential issues with computer systems

- Introducing and integrating new technologies into existing data-center environments
- Performing routine audits of systems and software
- Performing backups
- Applying operating system updates, patches, and configuration changes
- Installing and configuring new hardware and software
- Adding, removing, or updating user account information; resetting passwords, etc.
- Answering technical queries and assisting users
- Ensuring security
- Documenting the configuration of the system
- Troubleshooting any reported problems
- Optimizing system performance
- Ensuring that the network infrastructure is up and running
- Configuring, adding, and deleting file systems
- Maintaining knowledge of volume management tools like Veritas (now Symantec), Solaris ZFS, LVM. [7]

This scope in military networks also must include operational reporting, which allows for Command & Control (C2) in NetOps.

3.1.3 Navy Network Organization and Automation

Navy has never had a standardized base-line design for any global network infrastructure. It has instead been a patchwork of integration on multiple legacy systems that migrated over to support multiple micro architecture designs. Each micro architecture was designed as its own solution to a specific mission set and many times was not intended to be incorporated into new mission sets. Examples of these patchwork micro architectures will be described via use-cases in Chapter 4.

As a result of patchwork architecture, many NetOps actions are performed in a decentralized mode of operation, causing multiple redundant operational tasks to be shared at all NENs and their corresponding NOCs. These actions are then relayed up to NNWC over a human interface. Over the years, there has been an attempt to merge some of the NOC processes, but never as there been a formal framework put in place. A potential option for a formal framework is addressed in Section 6.2.

The current state of the Navy NetOps capabilities is unorganized, and unable to be efficiently managed. The grey clouds in Figure 3.2 are all independently operated NENs (a full-page version of Figure 3.2 is included as Appendix D). The Integrated Navy Operations Support System (INOSS) framework describes this picture in the following way [8]: "The isolated and disjointed evolution of Operations Support Systems (OSSs) across the DoDIN-N have produced an array of overlapping and redundant capabilities that are plagued with scalability and interoperability issues for end-to-end service visibility, fault isolation and remediation, configuration and change management, and provisioning of resources."

Figure 3.2, referenced from the FCC INOSS framework, illustrates the current state of Navy networks. It shows how all the Naval Computer and Telecommunications Area Master Station (NCTAMS) and Naval Computer and Telecommunications Stations (NCTSs) currently have no common linkages that can be used to synchronize information. Additionally NNWC, the Navy's command charged with the responsibility to manage all of this, is not in the picture nor are Navy Program Executive Offices (PEOs) responsible for Programs of Record (PORs) who build and support systems/applications such as Consolidated Afloat Networks and Enterprise Service (CANES), Navy Cyber Situational Awareness (NCSA) or Enterprise Network Management System (ENMS).

There is a lack of understanding exactly how the Navy's network is architected today, and the enabling of automated processes is key to any future success in pro-active C2.

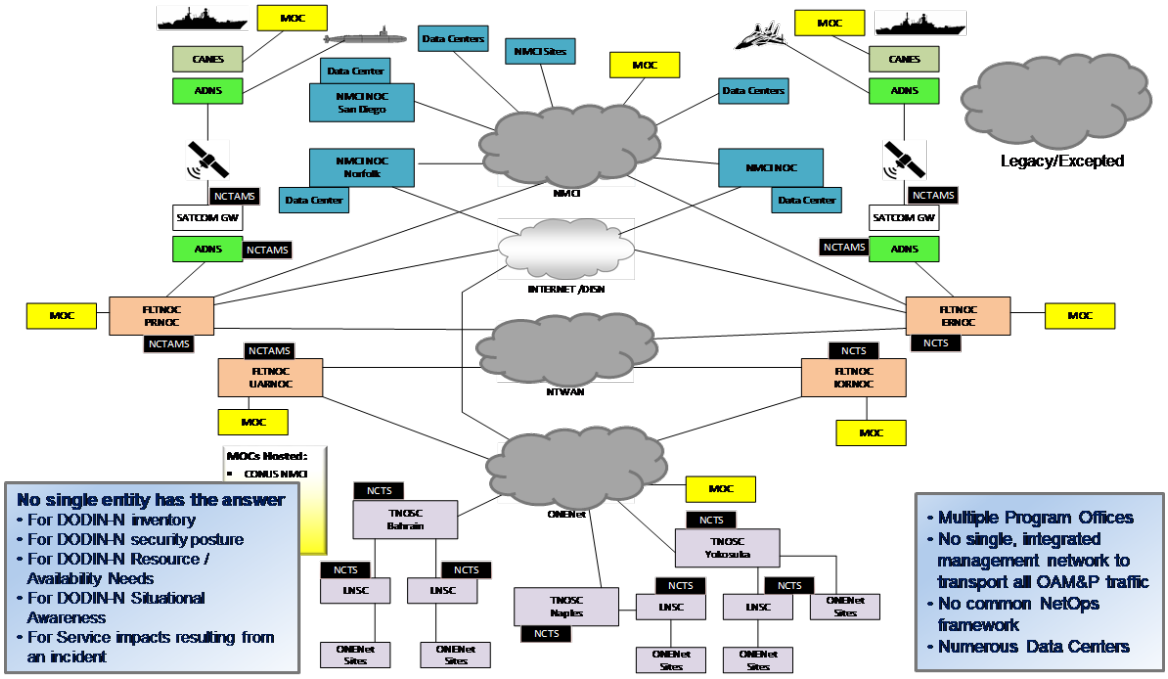


Figure 3.2. The Current State of Navy Networks. Source: [8]

3.2 The Concept of Development Operations

DevOps: A Software Architect's Perspective describes DevOps as the following: "DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality" [7].

The practice of DevOps in an organization means having the ability to rapidly integrate new developments in operational IT application. Section 3.1.2 lists the scope of technical employment a Network operator must manage, of those listed a DevOps team generally supports the following aspects of the elements automating a Network:

- Service desk operations. The service desk staff is responsible for handling all incidents and service requests and acts as first-level support for all problems.
- Image Technology experts. Ops typically has experts for networks, in-

formation security, storage, databases, internal servers, web servers and applications, and telephony.

- Image Day-to-day provisioning of IT services. These include periodic and repetitive maintenance operations, monitoring, backup, and facilities management. [7]

A team, which is referred to as a "DevOps team," is supported by the organization by enabling the infrastructure of a system to support key capabilities outlined in [7]. The ability of the system infrastructure to support these services are important elements in what is called a "deployment pipeline," the steps necessary for any developer committing new code into the production environment.

This infrastructure should support the following requirements:

- Team members can work on different versions of the system concurrently.
- Code developed by one team member does not overwrite the code developed by another team member by accident.
- Work is not lost if a team member suddenly leaves the team.
- Team members' code can be easily tested.
- Team members' code can be easily integrated with the code produced by other members of the same team.
- The code produced by one team can be easily integrated with code produced by other teams.
- An integrated version of the system can be easily deployed into various environments (e.g., testing, staging, and production).
- An integrated version of the system can be easily and fully tested without affecting the production version of the system.
- A recently deployed new version of the system can be closely supervised.
- Older versions of the code are available in case a problem develops once the code has been placed into production.

- Code can be rolled back in the case of a problem. [7]

Automation in NetOps depends on, "collecting events, detecting incidents, and measuring to determine if Service Level Agreements (SLAs) are being fulfilled..." in which "controls can be open-loop or closed-loop" [7]. Control of a system requires that monitoring capabilities be a part of the system. Monitoring enables detection of events, which can then have controls put in place which automate an operator's actions. Open-loop control is when a human operator is part of the automation process. For example, the creation of a template email that is automatically filled out with all technical information after a specific event allows for the operator to personalize prior to sending. Closed-loop control is when no human is part of the response in an automation process. An example would be if malicious packets were sent out from an end-point within the system and the closed-loop control automatically creates a firewall Access Control List (ACL) rule to prevent further packets from being sent and perhaps notifies the appropriate sub-systems without human intervention. Control processes can kick off other control processes, which enables as many automated features as needed. This combining of automation controls is called "orchestration."

The business logic where the line is drawn for between "dev" and "ops" in DevOps is defined in *DEVOPS: A Software Architects Perspective* is:

> The results of the monitoring are analyzed and acted upon by either the Dev or Ops group. One decision that must be made when instituting DevOps processes is: Which group is responsible for handling incidents?...One DevOps practice is to have the development group analyze the monitoring of the single system that they developed. Monitoring of multiple systems including the infrastructure will be the responsibility of the Ops group, which is also responsible for the escalation procedure for any incidents that require handling through one or more development teams. [7]

This book references multiple approaches that an organization can take to include developer/operator process integration. Points of interactions between "Dev" and "Ops" are described as software licensing issues and incident handling. Incident handling is where automation is best employed in NetOps.

A DevOps team for Navy Networks needs to be positioned within the organization close to the network operators who are responsible for the NetOps mission. In many cases a DevOps team member can be dual-hatted also as an operator because it is important for him/her to completely understand the technical operational requirements are needed when an automated control is necessary in a routine NetOps action. Additionally, a DevOps team member must understand when administrative actions are necessary in a given solution, as it cannot be employed without approved changes to the system's baseline configuration. In a nutshell, the DevOps team is a closely aligned team to operators doing NetOps that can then work together to reduce or eliminate human action redundancy and make changes to the system's baseline configuration.

A baseline configuration in the Navy is a state in which a system has received a certification to operate, otherwise known as an Authorization to Operate (ATO). The system is certified to operate by either an Fleet Authorization Official (FAO) or the Navy's Authorization Official (NAO), who are granted permission to accept risk to the network on behalf of FCC. FAOs are Systems Command (SYSCOM) entities such as Naval Sea Systems Command (NAVSEA), Naval Air Systems Command (NAVAIR), or Naval Information Warfare Command (NIWC). The NAO is an organization that oversees all ATO processes not connected to SYSCOMs. The standard auditing process the Navy employs to verify a systems compliance is called Risk Management Framework (RMF). RMF compliance would establish a baseline for the configuration and processes of a system to be deemed as having an "acceptable level of risk" by the operational commander, which would be FCC in the case of all NENs. When a system is found not operating within its baseline configuration, the Navy organization in charge of the system would fail an RMF audit. If justification for failure does not have substantial or appropriate mitigation, the organization would then be subject to refusal of Network connection services. If a system is refused service, that means the risk to the NetOps mission has been deemed greater than the value to a mission that the system supports. The system would then be taken offline until the risk has been mitigated to lower level, which would cause a loss in availability to the users of that system.

The DoD recently published an article on Development Security Operations (DEVSECOPS) titled *DOD Enterprise DevSecOps Reference Design*, which emphasizes the name switch from DevOps [9]. The difference between DevOps and DEVSECOPS is that "security" must always be present in the use of DevOps. While DoD has adopted the name "DEVSECOPS,"

this thesis will abbreviate this to just DevOps for brevity on the premise that security in development is always implied. A major point of the article was to consolidate the major points of realization which DevOps has brought to industry, and it provides a slew of rationale for the major switch by the DoD to adopt key practices. A great illustration provided from the article in Figure 3.3, shows the many challenges in incorporating DevOps into a military organization.

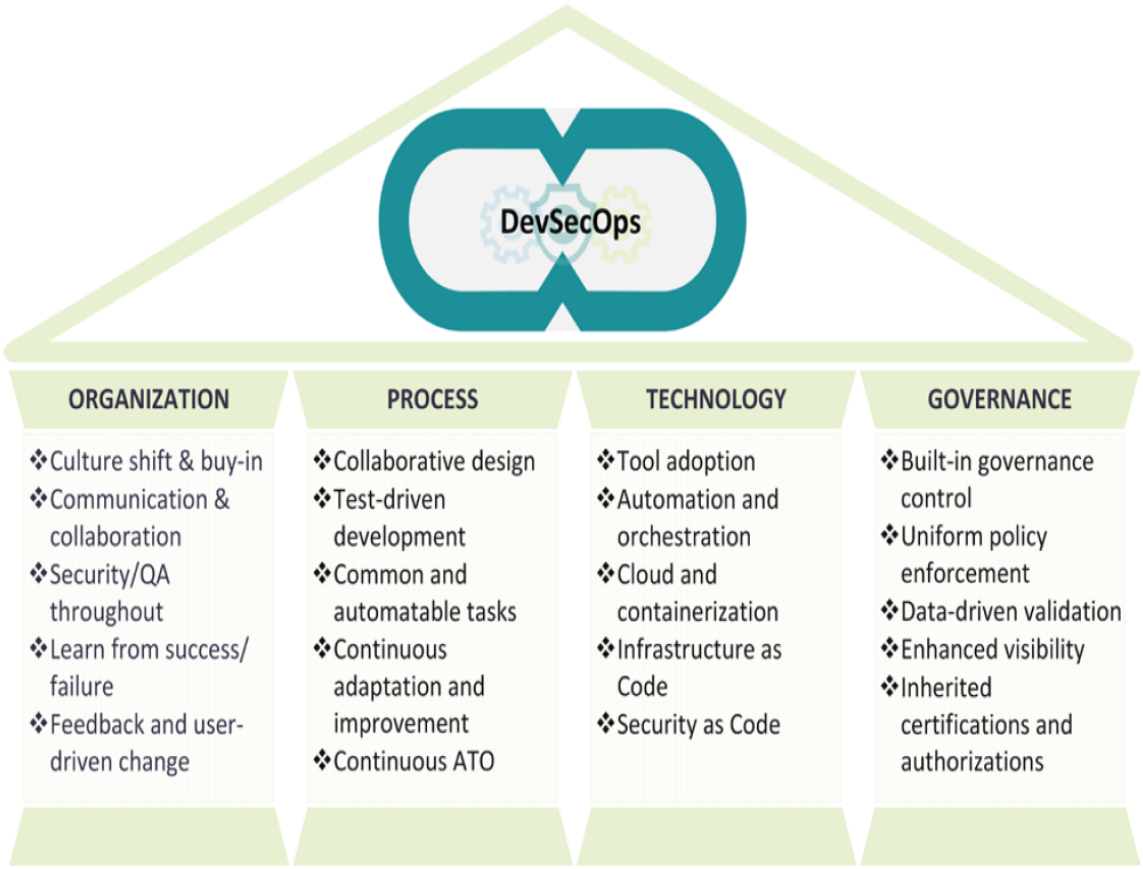


Figure 3.3. DevSecOps Pillars. Source: [9]

These pillars of DevOps incorporate all of the fundamental processes which must change in order for a military organization to fully embrace the concept. As this thesis focuses on the benefits in automation, some of the pillars of adoption will not be fully fleshed out in the solution presented in Chapter 6, and will require additional research in order for a NEN

to fully embrace the incorporation of DevOps; this is recommended for future research in Chapter 7.

The five fundamental best business practices are presented by Defense Threat Reduction Agency (DTRA)'s *Next-Generation Technology Governance* in *DOD Enterprise DevSecOps Reference Design*:

1. Run IT with Mission Discipline: Tie requirements back to your organization's mission. Every action should be aligned to the mission. If any are not, then an evaluation should be performed with Continuous Process Improvement to address how to tie actions to missions.
2. Invest in Automation: Automate everything possible, including actions, business processes, decisions, approvals, documentation, and more. Automation, including designs, interfaces, functional and security tests, and their related documentation, should create the Artifacts of Record that provide the body of evidence required by the Risk Management Framework (RMF) and for historical audits when needed.
3. Embrace Adaptability: Accept that change can be required at any time, and all options are available to achieve it. Fail fast, fail small, and fail forward. An example of failing forward is when a developer finds that a release does not work. Then instead of restoring the server to its predeployment state with the previous software, if the issue is simple enough the developer's change can fix it and address the issue through a newer release.
4. Promote Transparency: Offer open access across the organization to view the activities occurring within the automated process and to view the auto-generated Artifacts of Record. Transparency generates an environment for sharing ideas and developing solutions comprised of Subject Matter Expert (SME) or leads from across the enterprise in the form of cross-functional teams to avoid the "silo effect." When composed of all representative stakeholders, the team possesses the skills needed to build a mission system and the collective ingenuity necessary to overcome all encountered challenges.

5. Inherent Accountability: Push down or delegate responsibility to the lowest level:
    - Strategic: This is related to the Change Control Board (CCB) or Technical Review Board (TRB); it involves "Big Change" unstructured decisions. These infrequent and high-risk decisions have the potential to shape the strategy and mission of an organization.
    - Operational: (Various Scrum) Cross-cutting, semi-structured decisions. In these frequent and high-risk decisions, a series of small, interconnected decisions are made by different groups as part of a collaborative, end-to-end decision process.
    - Tactical: (Global Enterprise Partners (GEP)/Product Owner/Developers Activities) Delegated, structured decisions. These frequent and low-risk decisions are effectively handled by an individual or working team, with limited input from others. [9]

The document goes on to discuss potential for DoD supported automation framework with technical details, and it explains every step in the DevOps process. The illustration in Figure 3.4 illustrates the three main stages as "Plan", "Software Factory", and "Operations".

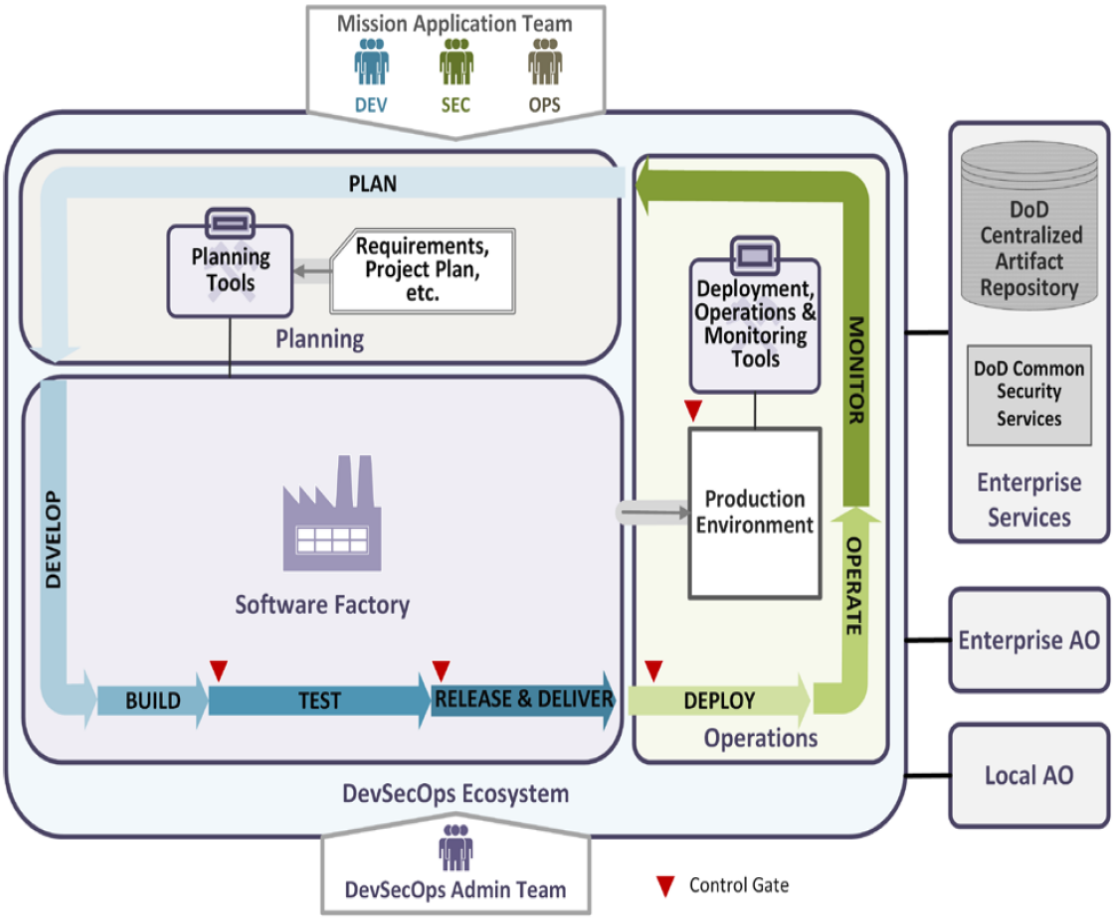


Figure 3.4. DevSecOps Ecosystem. Source: [9]

DTRA's *Next-Generation Technology Governance* explains that the "Plan" phase is intended to build and deliver a minimum viable product (MVP) for critical business needs, then begin a feedback loop as soon as possible to continue its development. The "Software Factory" phase is where automation tools are employed to produce new software otherwise known as the "deployment pipeline" as described earlier. There is mention that every environment is set up through the use of Infrastructure as Code (IAC) and Security as Code (SAC), which both would construct "virtual environments" which are used by developers. Virtual environments are a consistent tool that DevOps teams utilize for testing in the deployment pipeline and is part of the tool suite needed in a NEN infrastructure. The code is put into production and monitored closely in the last phase of Operations.

On the right of Figure 3.4 are the "External Systems" defined by the article as follows:

> The DevSecOps ecosystem itself and program applications depend on some DoD enterprise services to acquire the necessary baseline tools, application dependencies, and security services to operate.
>
> - DoD Centralized Artifact Repository (DCAR) holds the hardened VM images and hardened OCI compliant container images of: DevSecOps tools, container security tools, and common program platform components (e.g. COTS or open source products) that DoD program software teams can utilize as a baseline to facilitate the authorization process.
> - DoD Common Security Services are DoD enterprise-level common services that facilitate cybersecurity enforcement and IT management. One security service will perform traffic inspection and filtering to protect the mission enclave and mission applications. Some security service examples include firewalls; Intrusion Detection System (IDS)/Intrusion Prevention System (IPS); malware detection; data loss prevention; host-based security; log/telemetry aggregation and analysis; and Identity, Credential, and Access Management (ICAM). A Cybersecurity Service Provider (CSSP) will provide additional services, including Attack Sensing and Warning (ASW), Forensic Media Analysis (FMA), Assurance Vulnerability Management (AVM), Incident Reporting (IR), Incident Handling Response (IHR), Information Operations Condition (INFOCON), Cyber Protection Condition (CPCON), Malware Notification Protection (MNP), and Network Security Monitoring (NSM). [9]

The article makes the point that the DoD has given the Navy and all other services a well-defined way to attack DevOps, it still remains to decide on the tools for employment, external systems and to adopt this methodology into a framework. There is one point that this does not cover well, however. That is the physical proximity of the "Mission Application Team," or inter-communication methodologies of that team. The "Planning" phase tools vaguely mention the tool called the "Team Collaboration System," as a dependency in the activity of process of "software requirements analysis" [9].

Nonetheless there is no detail on how this facilitates feedback from an operator who uses the tools. This point is left open-ended in the document, perhaps to allow for organizations to adopt their own methodologies, or if it was overlooked, this is a gap in the suggested implementation of DEVSECOPS. Regardless, the long-term solution proposed in Section 6.4.1 recommends the use of "Onsite" DevOps at the NOC level.

According to [7], DevOps requires both automation and what essentially boils down to an empowerment of organized small sets of teams. Automation is performed through use of tools that enable the actions of the DevOps team, it explains the use of tools as:

> Tools can perform the actions required in each step of the process, check the validity of actions against the production environment or against some external specification, inform appropriate personnel of errors occurring in the process, and maintain a history of actions for quality control, reporting, and auditing purposes.
>
> Tools and scripts also can enforce organization-wide policies....
>
> Once tools become central to a set of processes, then the use of these tools must also be managed. Tools are invoked, for example, from scripts, configuration changes, or the operator's console. Where console commands are complicated, it is advisable to script their usage, even if there is only a handful of commands being used.... [7]

Automation tools within a NEN are later referred to as an "automation framework". Navy architecture today does not well support these types of tools.

According to [7], the goal of a DevOps team is to treat operators as "first-class" citizens. Once tools are made available, multiple teams working on micro-system automation with a team member called the "Service Owner" conducting coordinated actions is the best approach when it comes to utilization of DevOps teams. Small teams working alongside operators to support their mission are explained to be more effective in industry, largely in part because they employ process automations, reducing the systems requirement to have human operators, and also because having a small team effectively ensures team coordination can be deconflicted much faster with other teams focused on interfacing of other micro-systems.

3.2.1 NEN Barriers to DevOps

Even with an automation framework available, there are multiple barriers to DevOps that the Navy would need to overcome.

The largest barrier is the culture of operations. NEN operators' primary goals are to ensure minimum downtime, and high availability of the network. Avoiding change is a way to remove any cause of downtime, but the nature of developers is to encourage change and write new features into operation which may lead to down-time if code is faulty. Where the DevOps teams' job is to support the operators, they also need to be enabled by the NEN to make frequent incremental changes that can be tested and deployed into production rapidly.

Cost and displacement of personnel is another huge barrier to DevOps. The salary of a software-engineer is found to be 50% more than that of their system administrator counterpart [7] because a software-engineer must know most of the same information a system administrators does, and also understand the code enabling the interfaces they are operating. That level of knowledge enables the necessary automation of system administrator actions, which in turn would negate the purpose of a system administrator entirely displacing personnel from their assigned roles because the functions they perform would become automated. However, the NEN must first justify the investment in changing the DevOps support infrastructure and transitioning to higher paid personnel in DevOps. They also would then need to consider the need to lay-off or re-purpose their system administrators as DevOps automation increased. These shifts in workforce and money allocation could in turn disrupt the operation of a NEN in more ways than just support of a network's mission objectives.

Changes to the policy regarding baseline configuration change in the Navy would need to adopt DevOps practices into the process. The point of DevOps is to rapidly integrate new baseline configurations, which is not supported by today's antiquated software development life cycle (SDLC) [10].

These barriers to entry for DevOps must be addressed by the Navy and its corresponding NENs in order to empower a well-armed DevOps team.

3.3 Modern-Day NetOps Requires DevOps

One critical observation made after numerous Navy site visits is that none of the NENs visited employ any form of DevOps. The inconsistency of not having a DevOps team aligned with a NetOps team guarantees that all NetOps teams are reliant on manual network manipulation strategies to pass on information used to conduct C2 of the network.

Modern day operators cannot keep up with network demand without employing modern day practices in the conduct of NetOps. As technology continues to improve, so does the speed with which our operators have to respond to the demand placed on a network because many, if not all, situations in combat utilize a network and it is a fundamental part of the execution of the communication capabilities. The job of the DevOps team is to clearly understand any problems that arise within network operational capabilities and adapt a solution as fast as possible when no possible prepackaged solution is available to the operator. In order to do so, the operator must be able to communicate the requirements clearly and succinctly to the DevOps team.

A DevOps team on site ensures constant adaptation to any situation that an operator cannot solve alone and requires a developer-level knowledge utilizing code-based solutions. Where no operator interface is flexible enough to respond, a DevOps team's expertise is paramount to driving requirements and correcting fundamental system baseline issues. A fast deployment test capability for doing application update integration would be essential for a DevOps team to ensure that developed solutions do not break integrated system functionalities. However, fast deployment testing must be a feature provided by the PORs for the network.

A 2018 survey conducted by RedHat across multiple organizations suggests that the frequency of DevOps actions is actually higher than that of actions being taken by operators [11]. This suggests that employment of DevOps in commercial companies is actually more fundamental than of operators conducting NetOps, and it underscores the urgency for Navy networks to upgrade architectures. The business practices to keep pace with current technology of commercial networking peers, not to mention adversarial military networks need to be adopted by Navy networks.

3.4 Tools in DevOps

The Navy today employs old legacy systems which generally have no interoperable features allowing system administrators to access underlying data stored inside a given system. Without interoperability, this inhibits the ease of employment of technical solutions that enable data analysis. This has caused either no improvement in Navy data analysis or "hackish" quick fixes that have questionable security practices.

In order for DevOps teams to be successful, they require a technological update of accessible tools and, of course, system account privileges. NENs must undergo a transition to adopt the idea of scaling automation via network operation tools in order for the Navy to better enable DevOps teams. RedHat outlines how this can be done with these key tools and practices in a white paper titled *NetOps Meets DevOps* published in 2018:

- Shifting from command line to declarative, API driven interfaces
- Replacement of traditional, proprietary, centralized vendor specific tools to inclusive, multi-vendor orchestration systems
- Adoption of declarative, intent-driven automation and industry best practices instead of highly customized, brittle scripts and runbooks
- Use of policy driven automation vs hard coding rules into network systems. [11]

CHAPTER 4:
Problem Use-Cases

This chapter presents all the Navy use-cases found after multiple site visits to Navy network operational locations, which all have processes that need automation or have network architectural limitations that need to be addressed before automation can occur.

4.1 Use-Case Problems in NMCI and IT-21 NetOps

NMCI and IT-21 are the primary NENs focused on in this thesis, and this section identifies the autonomous capabilities missing when compared to technology of other well-established commercial networks. ONE-NET was not part of the focus of this study; however, it is the case that many of the problems found in these two NENs could utilize a shared solution with ONE-NET. This section first describes a few use-cases where autonomous capabilities can be utilized inside NEN processes, and the total amount of man-power that is needed to actually accomplish each process employed. It then touches on a couple systematic POR issues discovered after site visits.

4.2 Perspecta's Request Operations Center Use-Case

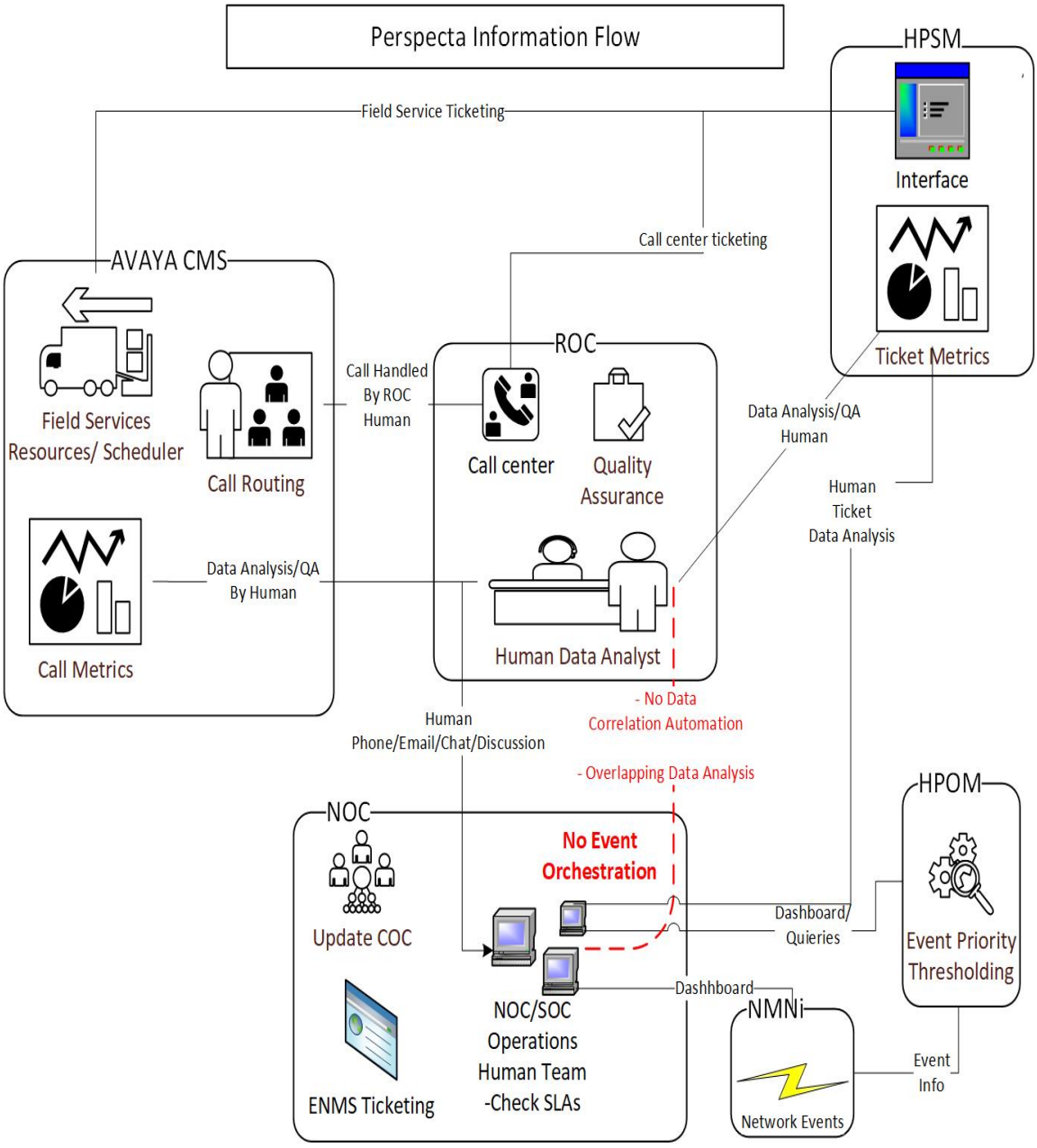
The NMCI NEN, which is currently run by a company called Perspecta, has a call center that must operate 50 shifts of 24-hour rotating staff of over 100 people, whose primary job is to answer the phone and create issue tickets, then try to resolve some of those issues by taking a caller through a multitude of different processes updated and maintained by a tool called Hewlett Packard Service Manager (HPSM). The annual cost of these tasks is likely a significant chunk of the NEN's operational budget.

In this monolithic system, issue resolutions get archived in HPSM never to be looked at again unless it is flagged by a Quality Assurance (QA) team member. The metrics for resolutions and call volume are recorded by another proprietary system called AVAYA Call Management System (CMS). AVAYA CMS has the capability to display charts, explain some of the call metrics, and is used to give Perspecta request operations center (ROC) operators insights on call analytics. However, all of this information is strictly contained

within the AVAYA software, and the data cannot be exported to an external system to be used for data correlations with NOC or Security Operations Center (SOC) operators. As a result, this is a system of human communication and coordination, where ROC operators must email or call NOC or SOC members about issues, and the data is limited to only the functionality of AVAYA.

All of the data contained within AVAYA and unused within HPSM could be utilized to reduce call-volume in the immediate future by having NOC operators access it for analysis and correlation to networking events. In the case of DevOps (if any existed), this information, when accessible, could be given automation controls that would allow for orchestration in the system if correlations to the data from multiple sources are found to link back to events. Currently, there is need for a gross amount of redundant human interaction to operate in the network after events occur. If call data were accessible and correlations were worked on, it could prove valuable to the NOC as the ROC could focus ensuring accuracy of call information, and the NOC would concurrently be working on issues which are trending to have wide-area correlations. This type of machine-teaming would allow for less redundant communications between the two teams and the adoption of a pro-active approach to solving potential problems before and event occurs.

The goal of the ROC system should always be to reduce future call volume. Reducing the call volume decreases the number of people needed to answer the phone, thereby making the system ultimately cost effective. Let us say that is the main goal of Perspecta's ROC. If classifiable problems are categorized, correlated, inferred, and the root cause is understood, then solving them outright will mitigate all future calls of that nature. Perspecta has employed humans to do these tasks, but those functions can be automated to a major extent. The red dotted line in Figure 4.1 represents the missing automated functions in doing root cause analysis. The figure also notes a lack of event orchestration and shows how the NOC team of people are doing the same analysis as the ROC team of people in terms of the HPSM ticketing system, which is a redundant use of man power.



The red dotted line depicts where there is an explicit lack of event orchestration

Figure 4.1. Perspecta's Interpreted Work-flow for Information and Events

Allowing the entire NEN's system the flexibility to synchronize data is paramount for situational awareness in today's modern NEN. The information flow of AVAYA is not optimally maximizing its use of the data analysts use perform analysis, resulting in redundantly

looking at the same information multiple times. This system could be built to sniff out problems for the operator and allow the operator to work toward placement of automated solutions instead of forcing two different human analysts to communicate slowly before data correlations can be found.

4.3 NMCI Incident Reporting Flow

An event is a basic rudimentary notification configured by a network device/application told to report when something out of the ordinary occurs. Events can be very minuscule notifications such as a small amount of lag on device/application, a larger issue like a notification that a device is no longer operating, or an automatic scan that revealed malware. There are so many events on networks that systems must have a way to filter out events that by themselves are not actionable information. In order to do this, events are many-to-one correlated to an "alert".

When there is an alert, that means one or more events met some pre-defined algorithmic risk threshold. The Navy defines those "thresholds" as Commander's Critical Informational Requirements (CCIRs) and SLAs. Perspecta must log tickets in a Navy system called ENMS when CCIRs or SLAs is triggered by an alert. "Alerts" are defined by both CCIRs and SLAs, thus events must be efficiently correlated in order to know when there needs to be an alert. An incident is one level above an alert as it is a prioritized alert where the need to address the cause of the alert is elevated. Various business processes that both Navy and Perspecta employ define which alerts become incidents. However, most alerts defined by CCIRs are considered incidents. It is, therefore, important to know and understand the flow of events through a system, a basic knowledge of the underlying tools employed, and how Perspecta correlates its events into alerts and incidents to gain insight into how the work-flow is currently being handled.

Hewlett Packard Enterprise (HPE)'s Network Node Manager version i (NNMi) is a proprietary network status Common Operational Picture (COP). Its job is to tap network infrastructure and provide outer/inner router information, Intrusion Prevention System (IPS) information, and perform "network node monitoring". This is the main tool that allows analysis of information to scan events. All the information is gathered and placed into a COP like display where it can be sorted and filtered. Granted, this is not a true "COP" by

military standards as it is not being correlated to a Navy operation. It can at best be thought of as a list of specific equipment statuses without correlation to the network's operational picture. Eventually this gap should be filled.

Hewlett Packard (HP) Network Node Manager i Smart Plug-in (ISPi) is a proprietary plug that extends the capability of NNMi. According to HP's tool documentation, ISPi provides the ability for data-collection thresholding capabilities for "bytes in, bytes out, buffers...etc." and also allows users to generate reports for "executive, overview, heat charts, top ten...etc." [12] This tool is used by Perspecta primarily for troubleshooting problems after they have occurred, and to generate 90 day Internet Control Message Protocol (ICMP) on round-trip reports for NNWC. In other words, ISPi is a plug-in to NNMi that gives operators the ability to do data correlation.

HPSM is a proprietary Perspecta ticketing tool (similar to Navy's ENMS) that has some automation capability built in to correlate events and create alerts for Perspecta identified cases of events found in NNMi. The alert criteria (algorithms) currently have a slow manual process for an operator to add customized alerts based on information from NNMi or ISPi. Without sending an email or making a phone call, there is no other way for the operator to request customized alerting behavior to be added. For work-flow of information operators uses NNMi/ISPi to observe events, and the correlation process (a repeatable process) needs a human to log a manual alert into HPSM when necessary. The automation capability between HPSM and NNMi is not routinely updated, nor shows any operator have any awareness of updating the event correlation automation processes between the two systems. In other words, there is a critical gap in the automation deployment process.

Enterprise Content Management & Delivery (ECMD) is yet another proprietary ticketing tool, used for escalation of tickets when there are incidents (prioritized alerts). This prioritization follows static rules that are defined by Perspecta's team detached from its NetOps teams. It is meant to show higher-up management the incidents that are supposedly being dealt with, and are based on Next Generation Enterprise Networks Re-compete (NGEN-R) contracted Service Level Requirements (SLRs) or FCC CCIRs. The kicker is, the information for this tool is always 24 hours behind. NNWC's Battle Watch Captain (BWC) has access to view this information via a web interface. Although this tool in particular can be used for trend analysis, it generally gets updated too slowly for use by

an operator, and it ensures upper management is viewing potentially outdated information. ENMS, the Navy's ticketing system, is updated manually by Perspecta via a web-interface when CCIRs are noticed to have been tripped, or when SLRs are noticeably not met.

NNWC's BWC must access and view all of these views in order to ascertain situational awareness. That means they must be tracking views for ENMS (East and West), HPSM and ECMD. NNWC merges all ENMS tickets under the East view to consolidate tickets from East and West into one view. NNWC's BWC then uses this information to conduct NetOps actions. It is little wonder why manual human communications is the primary method for notifications, in the operation of NMCI.

The way in which NNWC's BWC would self-ascertain situational awareness of NMCI is they would first compile information from 3 different locations. Next, they would filter out all redundant information. Then, in order to continuously monitor any one situation, they would ensure adequate tracking of all alerts are followed up on in a timely manner. If alerts have had no status update after a period of time, they would then ping the responsible NOC in charge for a new update. Seems straightforward; however, the NNWC's BWC has to do this not only for one network but correlate information across two additional NENs. This method of information flow necessitates slow manual laborious processes for NNWC and drives a need for NNWC to have its own process to correlate information which should have already been correlated by the NEN it came from.

The information management process at Perspecta needs to be re-tooled to empower its operators to effectively report incidents up the operational chain of command without the need for making a phone call or sending an email. If a human operator has to do the job of correlating the events before discovery of incidents, that means there is a lack of a process for automation.

All watch-standers on the Perspecta NMCI watch-floor do not have an ability or a process in place to pragmatically correct repeatable incidents identified and must utilize email, phone and/or message traffic methods to notify the operational chain of command (see Figure 4.2). All tickets must be updated by a manned machine interface. When there are multiple ticketing systems employed, this ensures a duplication of efforts between systems.

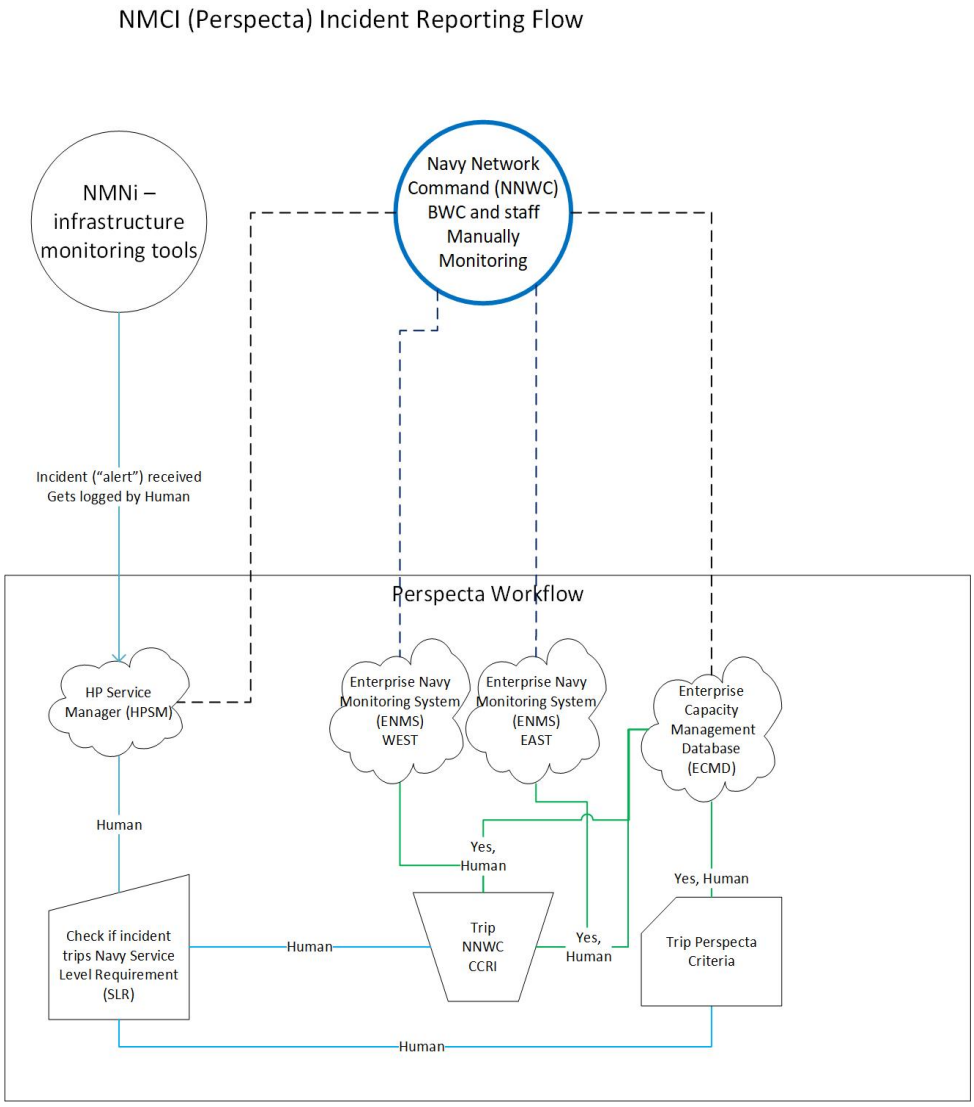


Figure 4.2. NMCI’s Tools Work-flow to Handle Incidents

Issues with Incident Reporting Summarized:

1. Manned machine interfaces for every ticket created.
2. Multiple ticketing systems with separate access controls.
3. Incident correlations are being done by the manned machine interface, with no effort to improve the process.

4. Operational reporting, is primarily email, phone, or message traffic despite all the other systems employed for escalation and ticket management

4.4 IT-21 Message Traffic System Use-Cases

After only a few site visits in conducting this research, four use-cases were identified to have automation inefficiencies in IT-21. This section will address the Navy's message traffic system inefficiencies for NOVA and PLA creation, and over-reliance on messages traffic for subsystem troubleshooting in ENMS and MUOS SATCOM troubleshooting.

4.4.1 NOVA is Not All That Automated

The NCTAMS Message Center is responsible for the operation of five message delivery systems (for military message traffic) and a couple of systems used to help manage. The messaging systems being used today are C2 Official Information Exchange (C2OIX), Fleet Message Exchange (FMX), Submarine Satellite Information Exchange Subsystem (SSIXS), Fleet Broadcast, and Common User Digital Information Exchange Subsystem (CUDIXS). Systems that are used to help manage these message systems are Master Update Authority (MUA) and the automated message store and forward system (NOVA).

NOVA is the overarching management system, used to help ensure C2OIX, FMX, Fleet Broadcast, and CUDIXS have a "green" status and stay operational. When either system gets an issue, NOVA gives no indication of what the issue is, it just shows a "red" status, and that only means that the system has a circuit down. NOVA is a simple system which operates off the packet routing indicator and other than displaying the information semi-intuitively, it does nothing else. Trouble-shooting of the system is then done using a human interface, and may include a soft crypto reset (bump-check) of the system, calling the tech-control, submitting an ENMS ticket after 10 minutes down (tripping a CCIR), and lastly calling Defense Information Systems Agency (DISA). The SSIXS messaging system employs its own similar system to manage its status, which does not talk to NOVA and must be manually troubleshot and checked for system outages.

Each one of these messaging systems has a unique purpose for which they are tactically employed by the Navy. The redundancy of having message traffic systems is a good thing

for the most part, however the management of each of these systems is very manual and troublesome. The only layer of automation employed to verify that systems are working is through the use of NOVA, which is separate from management of SSIXS.

When systems are overloaded, or if there are issues other than "circuit is up or down" such as system slow buffering, or message not delivered for a specific reason such as bad addressal of the message, NOVA will need a human operator to troubleshoot the issues with no base-knowledge of the problem other than the obvious sign that the system is not working. Additionally, at each NCTAMS (due to high-volume) the standard way to handle a downed communications path is to wait until users call or email about specific problems because there are no good indicators for operators to work with and waiting for the user to communicate an issue is more efficient than pro-actively gathering information to troubleshoot. Also, priority is not assigned to message traffic issues, unless explicitly stated in a separate manual medium of communication from a stakeholder or system user such as phone or email. In case it is not abundantly clear, the processes employed for this system are in a constant state of reaction as there are no real events, or event correlation happening in this system. At first sign of partial system degradation or malfunctioning, no actions are happening at the NOC unless a complete outage or there are human email/phone communications first. Operators are trained to ensure that systems stay up, but have no understanding of any underlying conditions that led to a circuit outage or any other system issues.

If NOVA was built within an automation framework, DevOps capabilities would enable the system to synchronize data correlation and allow for further troubleshooting of problems and potentially automate a response. As NOVA is not built within an automation framework and has no built-in scripting abilities, even motivated watch standers do not have the ability automate repetitive troubleshooting actions. If automation tools were available there could also be analysis of other packet information such as the layer two encapsulations which get stripped by the router. This analysis could say where the message had just come from immediately, in combination with correlating DISA circuit information, (that the watch stander must go to a website to check) and shared knowledge from the Navy Cyber Situational Awareness (NCSA) (a system mentioned in Section 4.5). This analysis could also allow for analysts to learn more about how these systems interact and fail, discover other potential contributing factors, and package up the results to be presented to the operator for

better informed action taking. This would be a great first step to improve this program.

4.4.2 IT-21 Plain Language Address (PLA) Creation and Maintenance

A PLA is similar to an email address for the Navy and is utilized to get "message traffic" (a form of mail) from one command to another. Every command/unit/organization can request to have a PLA via a message to the Master Update Authority (MUA). Details of this process are provided in the Naval Telecommunications Procedures - 4 (NTP-4) [13], a restricted For Official Use Only (FOUO) Navy publication that describes the process entirely. The process cannot be outlined fully in this document due to the sensitive nature of the information.

The MUA program is utilized for the management of all Navy organization PLAs. MUA is a program that NCTAMS - Pacific (NCTAMS-PAC) and NCTAMS - Atlantic (NCTAMS-LANT) take turns operating by handing it off each month to be managed by the other. The system is not in any way cloud interfaced; thus, it must be backed up every so often and synchronized between NCTAMS-PAC and NCTAMS-LANT. This is one of the first problems identified with this program. In order stay relevant to the latest PLAs, there should be some sort of a shared infrastructure. The management of the MUA program should never have to worry about the integrity of the information as this can be easily architected to abstract that portion of this program away from the operators via automation.

Another problem identified with this PLA process is the MUA does not in any way talk to Office of the Chief of Naval Operations (OPNAV), which maintains the Standard Navy Distribution List (SNDL), the authoritative list that is supposed to contain the Navy's Administrative Chain of Command (ADCON) hierarchy with up-to-date PLAs for every Navy organization. It is an easily identifiable way for OPNAV to update their SNDL by utilizing the MUA program as every single Navy command must utilize the MUA program in order to maintain their PLA. Having a succinct and updated ADCON hierarchy would solve multiple other problems in the NetOps mission environment, as the Navy utilizes the ADCON for multiple C2 enforcement mechanisms it has employed. Some details on how the Navy's Vulnerability Remediation Asset Manager (VRAM) tool is employed by the Navy and is mentioned as an area of future research in Section 7.1.5.

Lastly, the only way in which the MUA program will update a PLA is via a message traffic request. A policy/limitation currently in practice by MUA is whenever an incorrectly

formatted message is received the message will be ignored without giving notification to the message sender. The sender must first notice the lack of actions and call the NCTAMS in order to further inquire and correct the message. This is a small but blatantly obvious problem with the system. The system could be built to be self-corrective (automate a failure message to the recipient) instead of wasting man hours on waiting for a response, then calling to communicate the obvious lack of engagement when no action is taken.

A potential for automation between the communication system each organization is apparent. These poor business practices need to be smoothed out prior to being able to implement the automation's mentioned. Even given a framework to automate PLA processes, some cross-organizational process agreements must be established (such as OPNAV and the NCTAMS) and suggests a need for a Navy DevOps team to persistently identify and solve these types of issues.

4.4.3 IT-21 Redundancy in Communications Spot Report (COMSPOT) and ENMS Ticketing

ENMS is a multi-purpose system which was originally built to provide the Navy ticketing solutions but later scaled out to become an all-in-one informational dashboard display. ENMS was built by a company called BMC Software utilizing a commercial off the shelf (COTS) software called Remedy, popular for its use as an issue ticket tracking service. Remedy has gone through multiple generations of upgrades by BMC since this product was first installed, and the Navy is currently using a ten-year-old version of the software. ENMS, in addition to the ticketing support, also provides two web-interface tabs custom developed by the Navy POR PMW 790. These interfaces called Afloat Situational Awareness (SA) and Ashore SA, built for operators to see basic metrics about the flow of information through the network infrastructure to give a real-time status of ship communications to and from networking devices. The type of information provided to the system is network centric and could be anything from email back-logs to circuit status designations (up/down).

Due to a lack of training for the operator and poor operationally mappings of network metrics to mission functions, these custom built ENMS interfaces are not well understood by operators with access to the information. However, it is obvious that the software engineers from PMW 790 know the purpose of the information on these displays because

they understand the network infrastructure information. A Navy operator generally has a good working knowledge of the system and its mission, but is never as experienced as the developers who built or maintain the system. The translations of mission critical issues such as email back log, latency lag across specific components in a network, or storage capacities are near full, should be apparent before a phone call is necessary to inform the NOC.

System updates can be very slow to gather requirements, and the process for updating future iterations of the software needs to better account for the operator, and automation of the system. Later builds of ENMS often are scoped by Casualty Reports (CASREPs). A CASREP is a message traffic report that generally requests assistance of a Technical Representative (TECHREP) from the POR. The TECHREP then goes on site to troubleshoot the cause of the issue, the speed of which is prioritized by the CASREP. If the TECHREP finds a solution to the problem and it is system code related, it may get fixed temporarily or a temporary workaround will be put in place. The TECHREP would then ideally note the requirement to ensure future iterations of the same problem in software releases will correct the systems functionality. This is obviously a slow method of adapting ENMS to change, and ultimately reduces the usefulness of ENMS functions for the operator.

ENMS was originally employed with a system called Trouble Management System (TMS), a system used to support ENMS by automating the ability to read through COMSPOTs (a type of Naval Message traffic). TMS pulls the relevant information from COMSPOTs to utilize that information in correlating ENMS tickets. As pointed out in *CHIPS Magazine* in 2008, "the fleet continues to rely on trouble reports, Communications Spot Report (COMSPOT) naval messages, as the only "official" method of reporting, tracking and collaborating on communications and network outages. This slow reactive method largely precludes the use of automation, data mining and metrics analysis for problem management and process improvement..." [14]. This 12 year old observation of COMSPOT automation issues is still relevant today because the system has not changed. The article also mentions a study from 2005 sponsored by NNWC of afloat and shore IT services to have poor management processes:

- Few documented or repeatable processes exist on shore or ships.
- Multiple groups work independently on the same issues; problem corre-

lation is manual.

- No common operational trouble reports or logs exist.
- Joint Fleet Telecommunications Operations Centers (JFTOC) rely on printed COMSPOT reports using a paper-stacking, color-highlighting priority scheme.
- Problem management, including: detection; investigation; escalation; coordination; root cause analysis; and prevention of recurrences, minimally exists.
- Traditional hierarchical organizational structure and ineffective prioritization create inefficient reliance on senior-level personnel for routine tasks.

An internal site survey at NCTAMS concurs with this assessment, acknowledging inconsistent processes that rely on manual procedures and disparate tools, including 27 different autonomous "databases" and 46 paper-based logs, each independently and redundantly tracking/reporting operational processes... [14].

TMS system cannot be described as robust; however, the way the system operates today (designed prior to 2008) needs an operator to associate an ENMS ticket number as a line-item on the COMSPOT after creation of the ENMS ticket. If that COMSPOT does not have an associated ticket, TMS cannot ensure that ENMS will get updated with a new ticket, nor will it ensure that other tickets already created (that might be correlated) are getting updated either.

Let us walk through a couple of thought experiments to flush out why this is a problem.

Suppose "Operator A" is the ticket originator, and "Operator B" is a separate upstream NOC operator whom is helping to troubleshoot a problem.

1. Operator A makes an ENMS ticket. They then must send a COMSPOT with the associated ENMS ticket number. The ENMS interface does not automatically send the COMSPOT on behalf of operator A.
    - If ENMS auto generated the COMSPOT, it would reduce the total amount of work for operator A and increase the accuracy of the report to operator B.

Automated checks could also decrease the possibility that the ENMS has a duplicate ticket made by some unknowing operator.

2. Operator A may not have access to a web interface due to an outage, so they cannot make an ENMS ticket. Operator A then sends a COMSPOT without an associated ENMS ticket which in turn causes operator B to have to create the ENMS ticket for operator A.

    - Why is there not a work flow after TMS where an ENMS ticket is created automatically after reading through this COMSPOT? This would reduce the amount of work and increase the accuracy of the report.

3. Operator A must create a ticket on ENMS to troubleshoot a problem, and Operator B is more than one-layer removed from Operator A's problem. ENMS could understand operator A's context and help in the troubleshooting during ticket creation.

    - For example, if the system has context awareness it could have access to the GPS coordinates of the ship to know if a unit is afloat in a certain area of operations. It then would know what path is being used for IP services, as in which NOC it is configured for satellite communications. If then there is a satellite issue which is causing up/down link errors in IP services for Operator A, and Operator B is at the NOC, not the Regional Satellite Communications (SATCOM) Support Center (RSSC) or Global SATCOM Support Center (GSSC) (which provides communications transport to the NOC) and Operator A has not confirmed (with ENMS) during ticket creation that the issue has been troubleshot to the NOC yet, when they go create a ticket on ENMS, the system could help to ensure that the ticket and COMSPOT be routed to the appropriate location. This could be done by automating a service to assist the operator in ticket creation. This would then ensure accurate information gets to the right location faster.

4. Operator A creates an ENMS ticket, and mistakenly sends a COMSPOT referencing an incorrect ENMS ticket. How many man hours gets wrapped up trying to figure out the issue? This require Operator A to call for clarification to Operator B or send several emails to clear up any miscommunications. Also, Operator B might prioritize the wrong ENMS ticket based off the severity of COMSPOT.

- If ENMS automatically sent the COMSPOT when the ticket was created, this type of issue could easily be avoided.

ENMS is currently not assisting operators to reduce data redundancy, and in many cases, it is adding more complexity to the entire process. These are simple observations every operator has already made themselves; however, operators do not have the means or skill-sets to place permanent solutions into the system to reduce these work redundancies. DevOps should provide permanent solutions for cases like these.

4.4.4 Mobile User Objective System (MUOS) Satellite Logs in IT-21

One job both NCTAMS-LANT and NCTAMS-PAC have is to manage the MUOS satellite communications system. At NCTAMS-PAC the operators in charge of this system on the watch floor are called the Joint Fleet Telecommunications Operations Center (JFTOC) also known as the "front table". The front table has access to pull all real-time event logs for MUOS. Their primary job is to troubleshoot MUOS as problems occur (re-active environment) as well as work to allocate adequate resources for customers (navy units) when the need arises, which is communicated by individual units via message traffic, email, or phone call. They must then verbally report to the JFTOC Watch Officer (WO), the head of watch floor operations, so it can be relayed up the Chain of Command (COC).

These MUOS event logs are never reviewed until a problem arises and there is need to track down where an issue occurred. Primary alerts are when communications go down, but generally those are not seen. The majority of issues are discovered via COMSPOT message traffic from fleet units. As COMSPOTs come in, the watch stander will create ENMS tickets manually in order to elevate issues.

This reactive approach to management of this vital satellite system ensures that potential failures in the system will always arise. MUOS logs have the potential to be outfitted with a deep-learning model that more than likely would catch issues prior to COMSPOT generation. Obviously, humans could never parse all of the information that is generated day to day, so there is a need for an autonomous system to do this correlation analysis. Another issue here is that COMSPOTs are a slow method of communication to receive unit information status. There could be a correlation analysis between the COMSPOTs and MUOS log files to finding issue causation. The Navy could then automate actions prior to

unit COMSPOTs and pro-actively work on the problem. This, in turn, would speed up the process if not remedy any situation before it occurs, enabling a proactive response prior to events ever occurring.

The process as it stands now is when communication issues are local to the unit, the NCTAMS does not know the information until the COMSPOT comes in. In other words, even when an issue is known, the NCTAMS does not assist in trouble-shooting the issue until contacted by the unit (re-active approach). When issues arise at the NCTAMS, the unit will not know until a COMSPOT is issued by the NCTAMS. All of these COMSPOTs are human generated, which delays processing. If automated, the information regarding outages could be near instantaneous. Also, MUOS event log information is not ingested and shared with NCSA. The JFTOC front table, therefore, must inform the JFTOC WO verbally when problems arise and must use human generated reports to share the pertaining information.

4.5 Program of Record Architectural Problems

Navy Cyber Situational Awareness (NCSA) is a POR managed by PMW 130. Its purpose is to provide situational intelligence and visualization to distributed commanders, where unit networking/application information can be unpeeled to get to the smallest of details. This, in effect, helps the organization/command in understanding a NetOps related issue, and allow individual units to make assessments on their mission impact.

Currently, NCSA does not provide information in a high-fidelity format at the level of detail that most commands need, mostly because NCSA does not have access to high quality data. Due to some political and technical issues, a majority of the time NCSA spends is on the acquisition of data feeds because it takes time for them to get commands to allow access to the data. In some cases, older network devices/software do not have any built-in capability to acquire data and require custom development in order to push the correct information over to the NCSA platform. For some systems the information being pushed needs to include telemetry data about the information the device/application sends. This is likely because the application/hardware that NCSA would like to monitor was never built with the requirement to send analytical data, so a work around for each data stream needs to be built in order to get to the level of detail needed in NetOps.

PMW 130 also has another POR working in parallel with NCSA called "SHARKCAGE". SHARKCAGE is a capability for a global Navy Defense Cyber Operation (DCO) enclave that enables:

1. Visibility from external Navy boundary to tactical edge, to include Platform IT (PIT) systems and networks, and
2. Monitoring and decision support within decision cycle of adversary rapid technology engineering and insertion. [15]

SHARKCAGE essentially is a secondary effort to gather different types of information, with an ultimate goal to help facilitate its visualization with NCSA tools. Ultimately, the goal of NCSA and SHARKCAGE is for any Navy Commander to be able to pull up a view of his/her command and understand any risk to mission/objective caused by network issues that the command has brought to his/her attention. Consequently, this in turn would help that commander to make an informed tactical decision.

Many of the specifics needed in both of these programs were found to be some of the infrastructural necessities for having platform data storage, all of which are scaling issues that PMW 130's programs are eventually going to face. Currently, PMW 130 does not have all the information it would like because of funding and policy issues. Nonetheless as the organization acquires more and more data, this work in information-storage-focused IT will eventually become a problem, and it will require PMW 130 to shift focus off its current mission and into the mission of adopting a scalable data storage infrastructure. This would need to happen in order to continue scaling their other programs and allow the on-boarding of more data. How storage currently works is the data being acquired is to be held by a very expensive Splunk (COTS software) data storage solution.

Splunk maintains the type of information storage meant to be fast access but comes at a huge upfront associated cost. Splunk charges for indexing of all new data, storage of the data, both which come with very high upfront costs, but tapers off as more data gets indexed and stored. A leader at PMW 130 and various software engineers in the organization all admitted during an on-site discussion that the price tag to contain indexed information by Splunk is limiting for the future unless Splunk becomes a full-scale solution for more than just PMW-130. Data storage and indexing services adopted Navy wide would significantly cut some

of the associated cost and make the solution more affordable. For an in-depth discussion on Splunk's cost model, see Section 5.2.2. This, however, has some disadvantages discussed in Section 6.1.2.

PMW 130's effort to gather up and retain large amounts of networking/application information is very appealing to other PORs. For instance, PMW 160, is in charge of a POR called CANES/Agile Core Services (ACS). CANES/ACS allows developers to create fast applications that should be able to be accessed easily by shipboard personnel. This program is one of the Navy's first attempts at doing DevOps. CANES/ACS is a major participant in a Navy program called Compile to Combat in 24 Hours (C2C24), a Navy mission focused on "modernizing the afloat end-to-end architecture and allowing the Navy to deploy new software capabilities in less than 24 hours" [16]. CANES/ACS will be Navy's next generation tactical afloat network that brings people, technology and processes together that streamline the end-to-end flow from code development to capability delivery [17].

As the CANES/ACS capabilities begin to scale up, it is only a matter of time before PMW 160 developers are going to want access to the information being hosted by PMW 130 in order to enhance any application's enrichment of information. Utilizing the latest Network information pushed out from critical systems into PMW 130 data warehouse, it is paramount not only to the mission which PMW 160's developers support but also to potentially enrich the information of PMW 130. There will likely be a two way exchange of this information, as information gathered by the war-fighter (where PMW 160's application is hosted) will more than likely be corrected/corroborated with PMW 130's information by correlation analysis of the application built out on the CANES/ACS platform. Thus, again, it expands the mission set of PMW 130 to provide timely application programming interface (API) services to allow applications access into their data warehouse.

The mission set of PMW 130 is now huge. While they are funded to support development of NCSA and SHARKCAGE, they must also address a completely separate mission to build infrastructure which includes these four main areas of work:

- Discover and build customized software to get data into a consolidated location.
- Negotiate with commands on policy/SLAs for commands whose data they store.
- Warehouse all information in a scalable, cost efficient, highly accessible fashion.

- Support access of the information into their data warehouse from other PORs via a maintained API.

This type of mission creep is one that could easily fail PMW 130's primary mission set. At the very least it should be viewed as a problem that degrades the quality of the deliverable they are tasked to provide to the Navy. Their job should be to only create the tools that correlate and report on the data which is available. This separate mission of gather, facilitate, and warehouse Navy data could be an entire other POR to be funded. All of the excess items listed are in excess of their primary mission and shift the focus away from a very hard to solve problem to multiple vary hard to solve problems, decreasing the effectiveness of PMW-130's mission.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
DevOps and Network Architecture Solve the Problem

The focus of this chapter is to establish that DevOps and a supporting architecture must be adopted in order to solve the issues surrounding NetOps. This is done by qualitative analysis of commercial organizations and some smaller Navy organizations which have already begun to formally/informally adopt DevOps processes into daily NetOps practices. The goal of this chapter to is establish the thesis that only two fundamental changes are required to bring the modern Navy NEN into an age of proactive NetOps employment, as stated in Section 1.2.

5.1 Industries That Employ DevOps

This section discusses the history of DevOps process development at AT&T and how the company came to adopt DevOps processes, and develop the tools it uses today. It will then explain a tool suite that exemplifies an automation framework used for the purpose of DevOps and sold by Extreme Networking.

5.1.1 AT&T's Solution to Scaling Automation

AT&T's network is unique because it is very old and has gone through multiple architectural changes that have enabled it to perform its mission as it kept up with new technologies in competition with other network providers. Similar to the Navy, AT&T has many "legacy" systems which are still used on its production server and are customer facing. They are slowly moving to a Software Defined Network (SDN) type environment where they will define the components that are virtually being operated on, as they transition the legacy systems over to the newest method of operating in today's networked environment.

AT&T's Network Hierarchy

AT&T divides its network management into a three-tier global network/center-based operations structure [18].

Tier I is the Global Network Operations Center (GNOC), which is its overarching command and control center to include incident and outage management. They receive input from the network that is processed through various algorithms, one example being its 'correlation engine' which on average produces 150 actionable items from 15 million daily network alerts [19]. This processed input is fed to the watch-stander on their terminals or projections.

Tier II is the advanced technical services located in Georgia that provides advanced technical support for the physical systems.

Tier III, the regional component, contains network and service application reliability centers. These centers are comprised of smaller sites that receive work requests from the other two tiers as directed. A tier III site may have multiple types of tier II assets under its scope of responsibility.

Feeding into the tier system are alarms from the physical systems that include IT operations, network service operations, mobility (Radio Access Network), IP backbone, transport, and voice switching. This structure is key to how AT&T performs its operational information roll up, and it ensures the GNOC stays well informed on day-to-day interactions or problems.

Full Spectrum Automation at AT&T

The GNOC appears to heavily rely on two methods for network automation. The first uses a program called Rules You Build Yourself (RYBY) (pronounced RUBY) to define and test actions. RYBY is in-house software used to manage automation over its non-SDNs, and it is employed by about 40% of the infrastructure today as they transition to SDN technology. SDN's use open source software for automated actions called Open Network Automation Platform (ONAP). AT&T created this software as open source in hopes that it would be adopted by other companies and strengthened by the open-source community. AT&T continues to support and contribute to its open development, and it is publicly available software for adoption. This means that the Navy could, if they choose to do so, adopt some of the same AT&T automation tools without any licensing fees. In fact ONAP just recently worked a deal with Microsoft to bring new third-party satellite service functions called Virtual Network Functions (VNFs) into a marketplace for the tool [20]. VNFs mimic "physical network functions, such as routers, switches, and firewalls... run on Virtual Machines (VMs) and can, in theory, be managed and orchestrated with zero touch

in an SDN. Those VNF, which represent the 'intelligence' behind the network, may be chained together to form a complete networking communication service." [18].

The big difference between RYBY and ONAP is that inside an SDN, ONAP can be used to completely change the architecture of the system and correct/normalize data in API, which in the past could only be corrected by hardware level implementations. RYBY and ONAP both provide the ability to automate the network, take pre-defined actions over the AT&T network by scripting the actions necessary to correct errors, enable the system to adjust capacity to fluctuating traffic, and even update firewall inputs. AT&T intends to be 100% SDN managed by 2020, which means they intend to phase out RYBY.

RYBY was originally a third-party software that AT&T purchased and adapted to its needs. AT&T modified it to work over various platform types to include switches and routers. The software uses a simple "if this, then that" style of cause and effect correlation of alerts, but AT&T was very clear in its description of the software that any rule that they wanted to deploy was tested within a simulated network prior to deployment. The software is not limited to simple network administration commands; it could produce actions that can inject synthetic packets and parse information in order to make decisions based on Federal Communications Commission rules (important for internet service providers), SLA based agreements, service (Tier II) class environments, and measured metrics.

Similar to the Navy, the ticketing automation software used by the watch floor was "Remedy", but the company is transitioning to "Service Now". These ticketing systems were fully realized and had the capability to automate phone bridge connections, emails, and other notification/message relay abilities like text messaging. The ticketing systems were also implemented to automatically send AT&T's equivalent of FCC/NNWC CCIR emails when events need to have critical attention paid to them. This automation-heavy implementation of basic ticketing software has allowed the company significant success in getting the right information to the right people in the shortest amount of time and does not require the same amount of man-power that the Navy currently uses to implement ticketing.

How AT&T Started Down the Path to Automation

Chuck Kershner, one of the main presenters at an AT&T conference this thesis group attended, had been the daily operations manager of the GNOC for many years. In response

to continued queries regarding automated actions he told the origin story of how the AT&T GNOC got to where it is today. AT&T started to transform its network automation practices in the 1990s when they did not have any such system in place, and everything they did was heavily augmented by a human interface, similar to how the Navy does it now. This worked for a time, but they realized that the growth of the network would outpace their ability to pay for people to administer it. Realizing that change on the scale needed to be successful in the domain was difficult, they took a small team of people and gave them a six-month period to solve the problem.

The team analyzed system alerts and data-stream information which AT&T had been relying on to make network decisions and found 100 different situations where issues could be given a potential probability with categorized alarms and system data. "System data" consists of the type of networking metrics that could be automatically gathered without humans in the loop. They do this now by sending synthetic packets to networking nodes that are programmed to have an expected response that provides the targeted metric.

All this initial analysis was done using Excel, and it resulted in the team finding not only common network problems but also hardware and trust issues. Chuck gave an example of a hardware issue found based on information crossover alerts, which they came to find meant that a specific hardware component needed to be replaced. This led to associated correction actions which they then automated using RYBY. This was a hard buy for their upper management to allow automation control over various aspects of the network, so they slowly implemented those changes over time after doing much testing in simulated environments. Eventually, leadership realized the importance of the project and gave it their full support. AT&T now employs a full-time lab to figure out new automated network actions by doing "predictive analysis" and testing over the network.

AT&T and DevOps

AT&T unknowingly invented DevOps and integrated the idea into its architecture before the idea of DevOps formally existed. When Chuck assembled the team to perform this analysis, it was the beginning of an unanticipated change to the culture of AT&T. The team initially utilized the tools available to them such Windows Excel spreadsheets. Through analysis of the AT&T infrastructure, this team discovered that automation needed to be integrated into the system design, or else AT&T would always be behind the curve doing re-active cleanup

of system malfunctions. This changed the stove-piped culture of top-down management from distrusting developers with rapid integration once the value automation was realized, and the adoption of automation then became an important part of its infrastructure leading to the development of RYBY and eventually the creation of ONAP.

AT&T can now do predictive analysis, and test its automated features over the network. This saved AT&T millions in lost revenue from missing SLAs, or losing customers because of unplanned outages, and was key for them in adoption of future technologies and future success of the company. This in turn has freed their staff to continue developing proactive creative automation tools, which they have been able to package and sell as products. These products include AT&T's "Orchestrator" tool and their "Threat Insights" service, which enable any organization to employ some of the same automation techniques used by AT&T. Most importantly, AT&T has the infrastructure to support and automation framework, and it enables the DevOps process allowing the company to prosper in leading customer and product support.

5.1.2 Extreme Networks Solution to Scaling Automation

Extreme Networks offers a full suite of network management tools including device management and automation, policy for both physical and cloud-based applications, analytic visualizations, and compliance auto-configurations with alerts [21].

After a visit to its campus, the biggest take-away from the experience that related to automation was the intent behind the tools they have developed. Their suite of tools provides a much richer management experience of the network, allowing administrators to set up device configuration and compliance, and network workflow automation as well as configuring edge switches/routers and storing telemetry data in a data-center. All of these functions can be performed from within a single application and managed by a team of network engineers.

Once deployed, the expectation is for the client to hook their suite of tools up to any software used to analyze the information. This can be done via the Graphical User Interface (GUI) provided by Extreme. Figure 5.1 gives a visual depiction of the integration.

The GUI allows the operators to eliminate human error and automatically adjust to changes

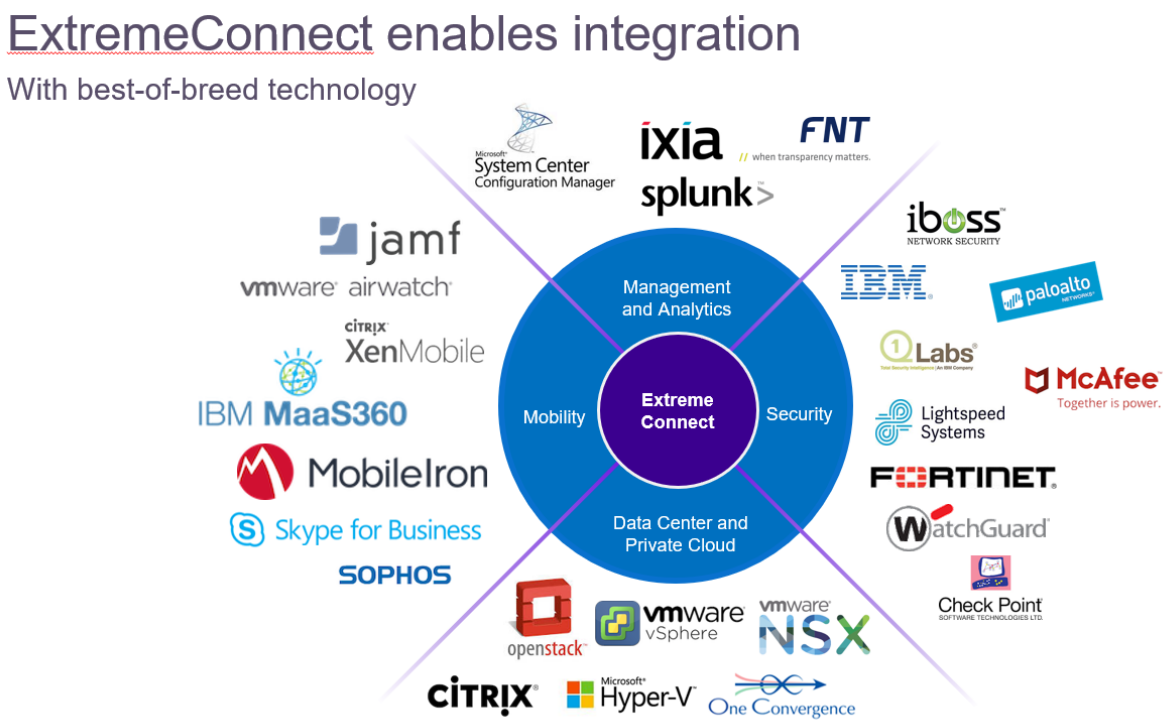


Figure 5.1. ExtremeConnect Integration with Multiple Tools. Source: [21]

saving time using an easy to manage interface called Extreme Management Center (XMC). XMC comes with built-in tools such as workflow builder, allowing users to manage flow of network traffic and configure alerts to be triggered. Another tool called workflow dashboard gives operators insight into the status of all workflows in a graphical representation and allows operators with the appropriate access to manage those workflows if there is need for a change using the GUI.

Note that in the description of DevOps tools in Section 3.4, Extreme is catering to the exact needs of an organization that employs DevOps. During their brief, they also mentioned the fact that DevOps was the exact target for which the tools were developed [21].

5.2 Navy Internal Developments

As AT&T and Extreme provide great examples as to how the employment of a good framework in combination with DevOps is trending with industry, this section will show use-case by NPS and an effort on part of Perspecta to begin the process of DevOps employment showing that Navy internal organizations are transitioning their processes to utilize DevOps processes. It will also touch on a few small development efforts in information pooling technologies that are starting to get built and utilized, yet they are not widespread for Navy-wide development.

5.2.1 NPS's Practices in Automation

NPS is considered an "Excepted Network" by the Navy. This categorization specifically applies to all networks that are not managed by the largest three networks described in Chapter 3. Although NPS's network is much more limited in scope and size in comparison to a NEN architecture, it still performs many of the same functions. NPS's architecture is less rigid than that of a NEN, so it has been able to employ DevOps controls in the use of its network's operation recently.

An example of a similar function performed is monitoring of security events. NPS utilizes a suite of software to automate actions on the behalf of the operator when events trigger at specified alerting thresholds. Because the team at NPS utilizes well-informed network process documentation, I have been able to outline a couple of practices that NENs could potentially adopt at a larger scale.

Process orchestration can be performed utilizing a tool called "Demisto". Note in Figure 5.2 NPS has been able to automate a single action to retrieve an email using version 2 of Exchange Web Server (EWS v2). A small automation flow like this can then be combined into a larger work-flow such as detecting spear-phishing emails, as Figure 5.3 demonstrates.

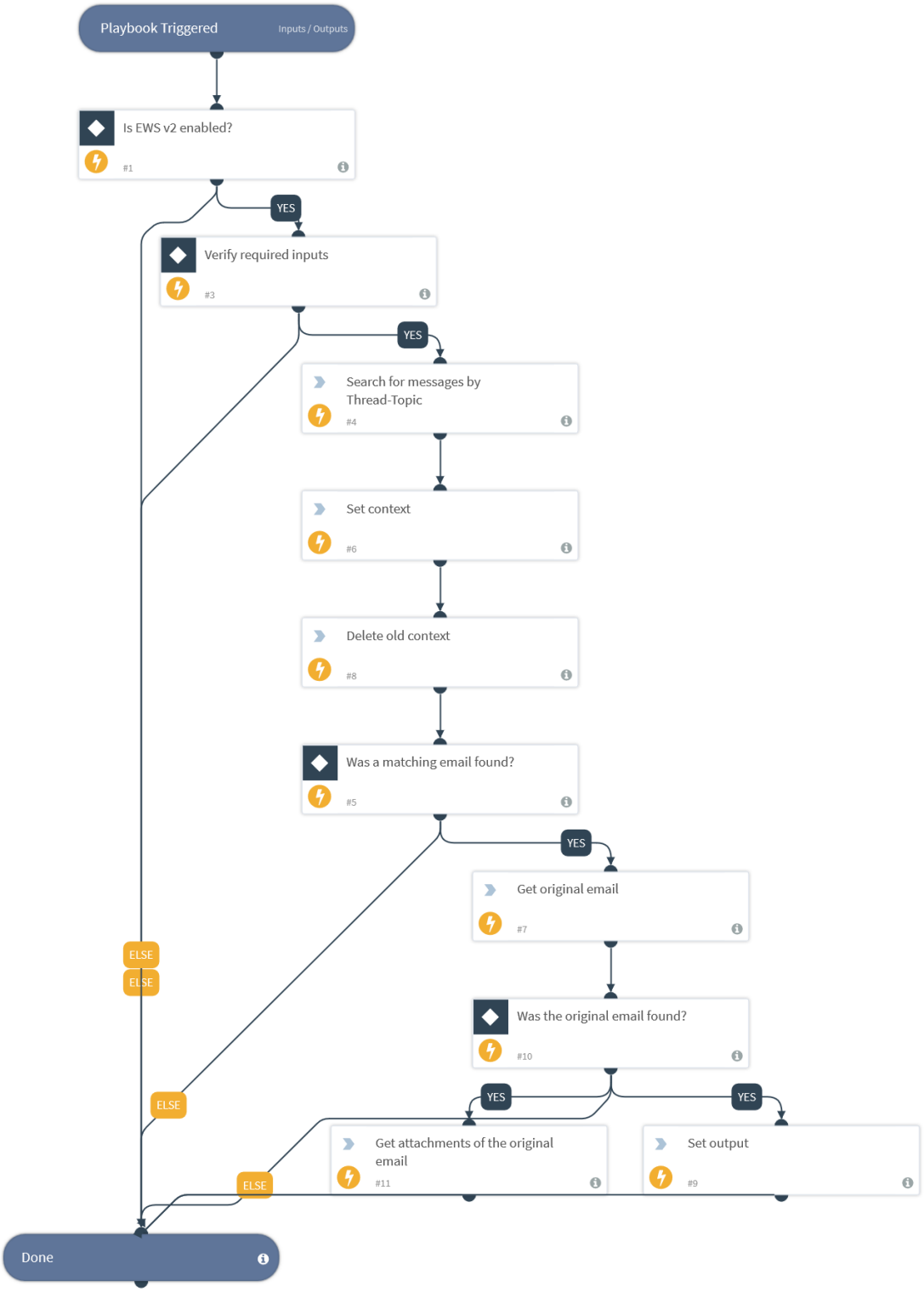
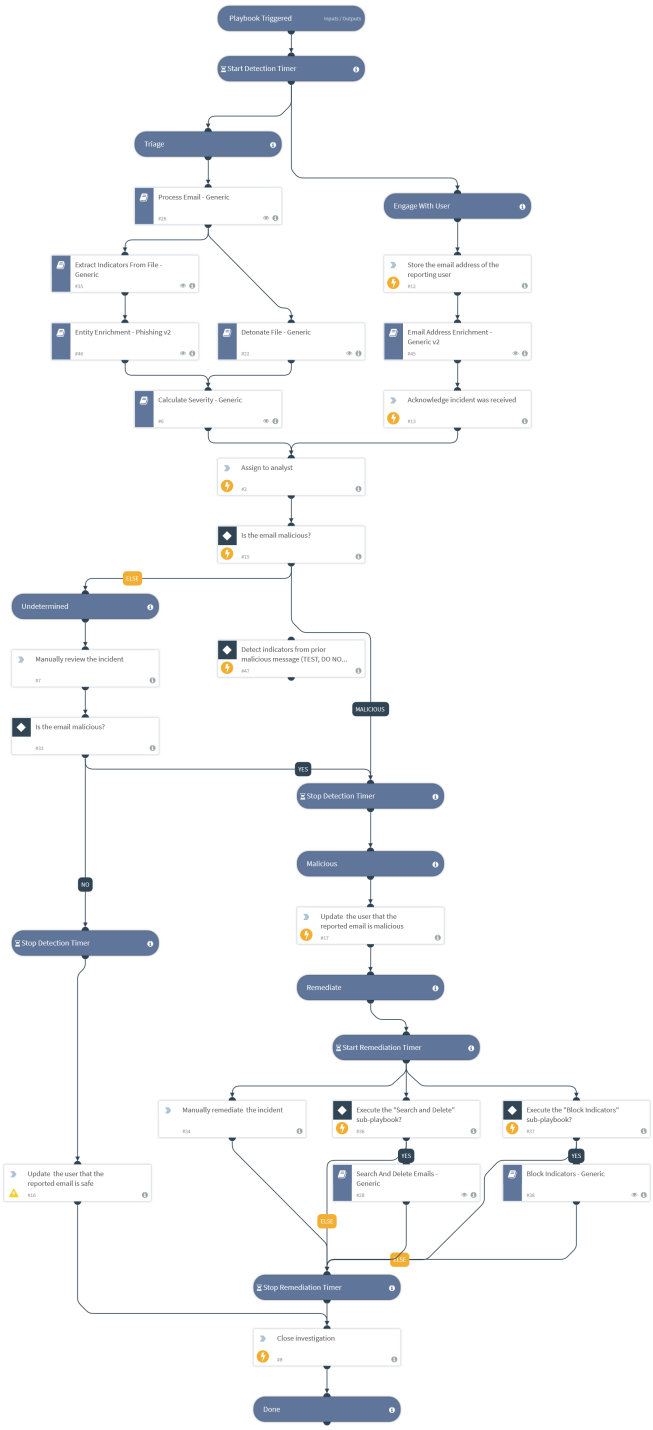


Figure 5.2. Example of Simple Email Event Orchestration



This example of complex event orchestration can utilize multiple other smaller orchestrations to assist in defensive actions associated to spear-phishing campaigns

Figure 5.3. Example of Complex Event Orchestration

These types of modern event orchestration controls are assisting the NPS operators with their job, by enabling them to continuously update and refine their automation processes. The work flow automation process has led NPS network teams to cultivate DevOps as part of their internal processes. Any automation built by an operator can be tested locally by the operator as it is developed. Then it is passed on to staffed developers who ensure the processes are adequately tested and ensure a smooth deployment onto the production environment.

Another product developed by its team is the documentation of a mature process. The documentation of this workflow is in Appendices A, B, and C. NPS documents the sophisticated workflow of functions employed by its operators, so that future operators can learn the process very easily by reading the documentation. When we asked for automation documentation, NOCs did not have any such type of product.

5.2.2 NPS's Solution for Data Storage

Lastly, NPS like all other NEN's has needed to put into place a data storage architecture. Similar to the way NCSA is trying to solve this problem, it has adopted the solutions offered by Splunk. After a discussion with one of their SOC operators as to the reasoning behind this choice, it came down to the optimized tooling provided by Splunk and a horizontally asymptotic scaling data cost model (Figure 5.4). Splunk employs its data cost scaling services by charging for "indexed" data. This cost model is unique in that Splunk gets paid for use of its software to organize data into a special format (another way to describe indexing). Data mostly consists of packets of information used to pass over the internet, or system logs on any device. Every time Splunk "indexes" a piece of data, the user pays a fee. Once data is indexed, the rest of Splunk software is used to correlate all the indexed data to create information that can be used to create events and alerts needed for automation practices. Indexed data is stored by NPS in servers that they maintain.

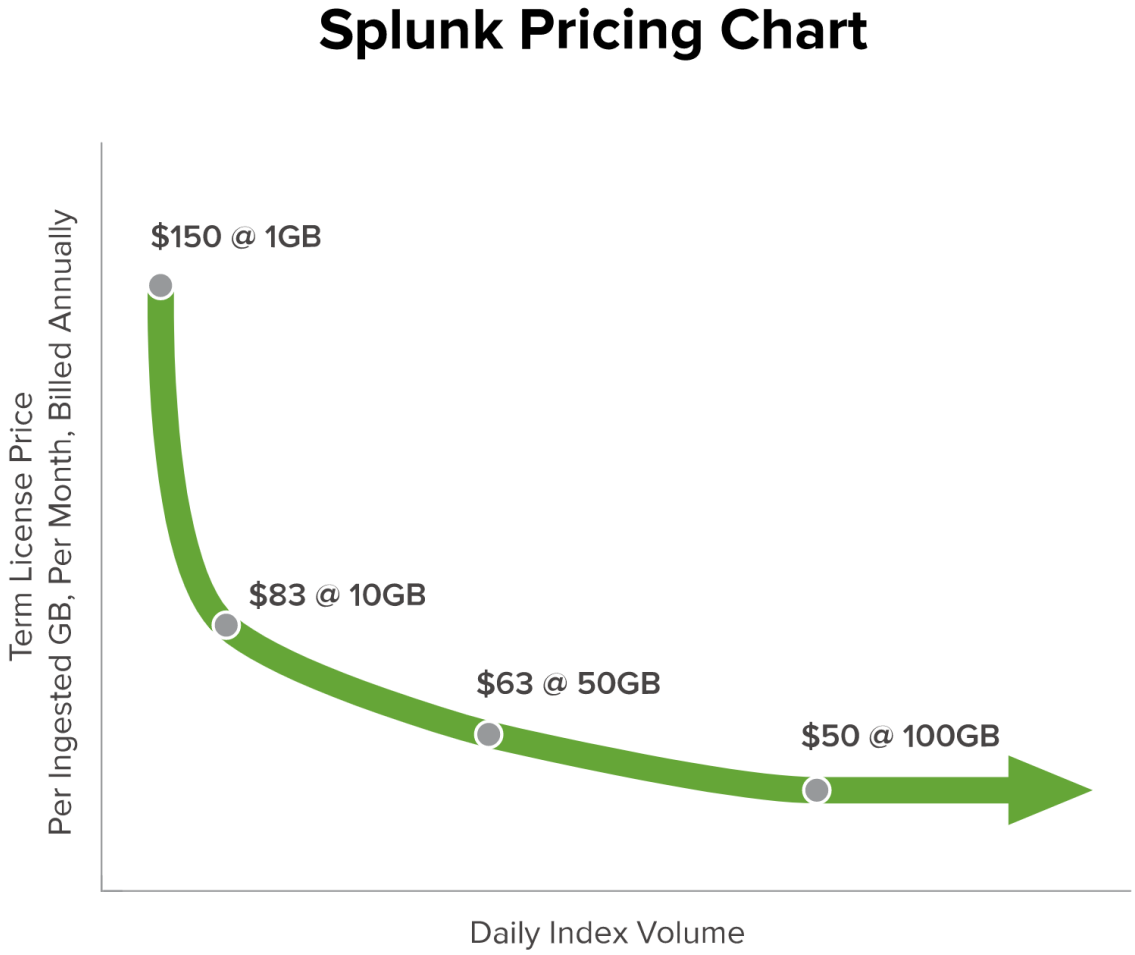


Figure 5.4. Splunk Horizontally Asymptotic Cost Model for Data Indexing. Source: [22]

This model allows NPS to deploy as many Splunk instances as it wants. This is a unique cost model compared to most software services, which make users pay per instance not per metric produced by service of the software instance. Since the number of devices added and removed from a network constantly changes, figuring out license agreements for each device to have software deployed on is more complicated than simply freely deploying a Splunk instance and paying for the data that eventually gets indexed.

As a large organization, NPS has a constant flow of data needing to be indexed, so the end deployment of Splunk becomes more cost effective as the need to index data goes up. Compared to a traditional cost model, where service is paid per instance, this means that the

price per additional service automatically becomes less as part of the cost model without need for re-negotiation or agreement for terms of licensing.

This brings up one key issue for the organization's employment of Splunk, which is the cost when abnormally high amounts of information needs to be indexed. This could happen when multiple users must stream training videos at the same time, due to abnormal end of the year job requirements, or even if a botnet infects the organization and data traffic peaks due to malware and unusual job environmental conditions. Splunk has addressed this issue by giving NPS a soft-limit on the indexing. This means that whenever NPS hits its soft-limit, all indexed data will continue to be indexed at no additional cost. Splunk will then re-negotiate soft-limits if overall traffic continues to average higher than prior soft-limit negotiated levels. This also means that if an organization grows before the term limit for the soft-limit re-negotiation, then that growth will be free up until re-negotiation of new soft-limit amounts.

5.2.3 Perspecta's Plan for DevOps

After a site visit in Norfolk with Perspecta (the company contracted to run NMCI), it was found that there was a huge push for a new paradigm of NetOps. Perspecta put fourth new plans in order to compete for the NGEN-R contract which would continue to enable Perspecta to be paid by the Navy to manage NMCI.

Perspecta has realized the need for DevOps. They have addressed implementation using a new plan that employs "Service Executives" to implement their vision of Perspecta's "DevOps Transformation" [23]. The problems presented in Section 4.2 regarding the use-case for doing analytics with call center data and reporting workflows in Section 4.3 may even be addressed by this new change in practices. Figure 5.5 outlines an interpreted graphical implementation of how Perspecta's executives will handle DevOps employment into the Perspecta NetOps work-flow.

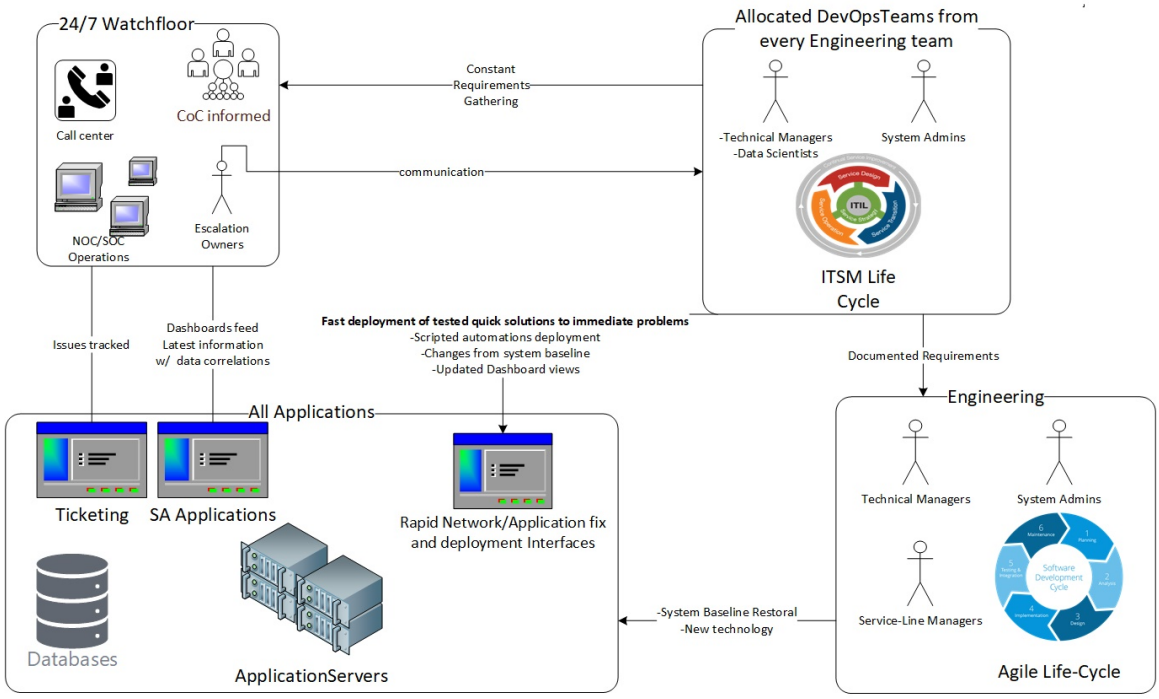


Figure 5.5. Perspecta's Interpreted DevOps Solution. Adapted from: [23]–[25]

What is unique to their implementation of DevOps is the employment of Agile engineering principles to the IT Service Management (ITSM) life cycle. Perspecta describes engineering teams who are tasked with continual improvement to the operational NetOps team will have "Escalation Owners" as part of the watch floor to seek out problems in the operational environment. They are tasked with giving feedback to a DevOps team who can make rapid changes as part of the normal ITSM practices but are highly technical because each come from one of the engineering teams tasked with Agile software development actions.

Regardless of whether or not this method of DevOps implementation is the best or not, it is important to note that Perspecta already understands the importance of DevOps and is actively pursuing a solution. Perspecta is doing this in order to meet its own SLAs without any hierarchical guidance or synchronization with the Navy for an Enterprise policy in the practice of DevOps.

5.2.4 Developments Made by NNWC for a Data-Lake

Big data has been identified as an issue for a long time in the Navy. In an article from CHIPS magazine in 2016, it was realized by Marine Capt. Benjamin Gallo that the Navy needs to figure out a solution to big data analytics that can reduce and consolidate redundant data repositories.

> The greatest challenge, however, will be moving from an ad hoc collection of individual data efforts to an optimized, enterprise level system. While this enterprise level effort has yet to be fully realized, unit level data projects often help to map the way ahead. Initiatives to map file content, reduce redundancy and obsolete files in data repositories, and connect legacy systems are underway in both the Navy and Marine Corps. These efforts will help shape the data landscape, save time and money, and provide valuable insights to upcoming efforts. [26]

This view has been all too true and there has been little movement in way of getting a scalable enterprise solution in place. The good news is that NNWC understands this problem and has taken steps in prototyping a potential solution for the entire Navy. There has not been much advertisement in regards to this movement as the solution has not yet had time to be fully implemented, but the fact that this effort exists supports the thesis's claim that a unified implementation of data repository architecture is a necessary component of big data analytical tools used in DevOps.

NNWC's job is to do NetOps efficiently and proactively. The roadblocks encountered during the execution of this mission set have been mostly access to data. They have employed a small team to begin the process of taking critical data-taps on Navy infrastructure, and finding a way to store the data needed for Operators to do data analysis at an appropriate classification level. This ongoing-effort to acquire data, unsurprisingly, requires the use of DevOps methodologies to ensure that network operators' requirements for data correlation can be met if sent to a data-lake solution. Jobs in which DevOps plays a role are figuring out what data gets stored in the data-lake, how that data needs to be indexed, and what type of correlations/functions need to be available for operators trying to extract information.

This effort is significant, because its mere existence alludes there are efforts at every level

to consolidate information. NNWC should, in all practicality, already be able to tap data from its NEN counterparts without the need to store the data itself, but because there is not a universal architecture in place for NEN storage, they have found that this is a necessary step in the process to glean insights from data analytics. If it needs to build this system out, then every system must have something similar to conduct a NetOps mission. Thus, the statement above is very valid, and little has been done in way of getting a scalable enterprise solution in place.

5.3 The Qualitative Proof

From the outset, the point of this thesis was to show that NENs could not be proactive in the conduct of NetOps without adopting two key changes.

Chapter 4 over-viewed multiple current use-cases where NENs and their associated PORs cannot be proactive. That chapter ultimately showed that today's Navy is not proactive in multiple areas of the NetOps mission-set.

This chapter has covered the many internal and commercial efforts to show how industry and small efforts within the Navy are setting out to solve specific problems they have identified themselves. There is no service level initiative for the internal solutions being developed, so they have just been adopted out of sheer necessity. This last point is important because the developed solutions as mentioned in quote from Section 5.2.4 the "ad hoc collection of individual data efforts," are completely independent progressions of Network architecture and those independent processes are coming to the same conclusions but implementing their solutions uniquely [26].

The first proposed change to adopt for proactive operations is to unify the architecture for data consolidation under a single Navy Organization per the following:

1. Shown in Section 5.2.4, NNWC's need for a data-lake to conduct NetOps extends to all sub-organizations confirming that the NNWC has already come to this realization and is making an effort to give a proof of concept on this point.

2. The need for an ability to access data freely is also clearly outlined in the use-case of Section 4.2 that covers the ROC (call center operations by Perspecta for NMCI) and

how data is saved in inaccessible proprietary subsystem.

3. It is further shown to be an issue for NMCI and NNWC in accessing event data in Section 4.3 where the primary mechanism for NNWC to get additional input on alerts is by emailing or calling the NOC.

4. This proposed change is also solidified in Section 4.5, where PMW 130 is building out their own uncoordinated data storage and analytics solution for NCSA and SHARKCAGE out of necessity. If another command was to take over the data, acquisition, storage and access jobs, this would drastically reduce the workload of PMW-130 or any other POR service, and free up time for mission specific focus. Granted, Splunk, in this specific situation, is in fact providing the storage and access services as a Navy external service. However, it would not solve how commands implement methods to push data up to Splunk.

5. Lastly, these points support the claims in [26] where the author suggests that a unified data solution should be a natural next step for the Navy. This would involve developing data models and an ontology for information related to network operations.

The second proposed change is to build into all software development lifecycles, a DevOps interface and environment that can quickly implement technical changes in all NEN architectures to facilitate faster automation otherwise known as an "automation framework". This proposed change is supported by the following:

1. Section 5.1.1 explains how AT&T arrived at the conclusion that as a large organization it could not keep up with its massive network demands without the use of DevOps. If AT&T needs DevOps, it is reasonable to conclude that the Navy must also require it.

2. Section 5.1.2 shows how modern tools developed by a commercial organization, designed for Network management and the NetOps mission set, are built specifically around the premise that DevOps is an integral component. The logic here is that if modern capabilities developed by today's industries are seen as a necessity for proper network operation, then that reasoning must also hold true for Navy networks as well.

3. Organizations internal to the Navy, such as NPS, are already implementing DevOps procedures to enhance their NetOps capabilities as described in Section 5.2.1. Additionally, Perspecta is shown in Section 5.2.3 to be developing its own DevOps initiatives to be competitive for the next NGEN-R contract. Similar to the last point for the first proposed change, these internal organizations are paving the way for Enterprise DevOps. This reasoning suggests that Navy needs to adopt a universal solution to the implementation and cultivation of coordinated DevOps, thus an automation framework needs to be put in place to enable the practice.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 6:
Navy Automation Recommendations

The previous chapter supports the hypothesis through qualitative analysis that shows why NEN architecture needs to adopt a unified data consolidation framework and showed why DevOps must also be adopted by NEN processes. The rest of this chapter is a proposal for two potential solutions.

This chapter suggests that NENs can implement those two proposed changes via a short and long-term strategy employment. The short-term solution utilizes Splunk for data-consolidation and explains how Splunk could help the Navy "almost" adopt the first change quickly, but it has some major disadvantages.

The long-term solution presented utilizes network architecture framework research already conducted by FCC called INOSS to adapt the current Navy network architecture and suggests the placement of DevOps into components of the framework. Since INOSS is a new concept, there are a few suggested modifications to that research that might help in the adoption of automation and incorporation of DevOps.

6.1 A Short-Term Solution to Data Role-up in NetOps

The observation taken from NPS is that the Splunk tools enable any operator who is trained in its use to analyze information and automate the process to create alerts. Installing Splunk seems to allow the organization to better see the gaps in that information and fill those gaps by ensuring its network configuration can provide details through the use of new methods or technology.

Data consolidation can be done in many ways. The key component for success is having a single tool that enables commands to improve automation gradually and to correlate information from multiple locations in a fast and efficient manner. As seen in Chapters 4 and 5 some Navy organizations have already semi-solved this this capability gap through the use of Splunk software. Splunk's method of indexing data and tools developed to correlate information for alerts are top notch. Their pricing model favors large scale deployment,

as it is easily scalable because its acquisition and adoption does not have to be integral to the architecture for which it is installed. Commands/PORs are currently migrating their network architectures to incorporate Splunk processes due to these reasons. Splunk can simply get the job done, so it makes sense as part of a short-term solution, and commands can adopt Splunk as their capability to consolidate data in this manner.

If a NEN was to adopt Splunk as its software of choice, to crawl and index their information, this would assist in the shift of its culture in the direction of DevOps employment in NetOps. The installment of Splunk instances is not a simple process. NPS is currently in the process itself of incorporating Splunk into its architecture, and it seems that the process will be continuous. First one must consider the organizations in which Splunk is employed, and how indexed data is processed, which means choosing servers to load balance multiple instances deployed across a region. This also means dealing with unique use-cases, such as sub-systems that are not always connected to a network (a use-case that NENs face). Second is ensuring that information that Splunk instances are providing matches the tasks of the network operators who use it. This requires that the team tasked to the deployment of Splunk are also operators, or working closely with their NetOps counter-parts.

A realization made through this process by NPS has been that it needed another tool called Forescout installed in the architecture, which ensured that end-clients provided Splunk the adequate information such as the device's operating system version and current application metrics. Forescout and Splunk work well together because Forescout can also manage the end-client's use, but the primary takeaway from this is that Splunk enabled NPS to see the gaps in information. This motivated a change in its architecture that naturally employs the use of DevOps in the process. The process continues because the architecture of NPS's network is always changing, as it must be able manage change that comes from the nature of cyber/network research. The DevOps team stays on top of these changes but continues to find new use cases that Splunk can pull additional information that needs to be correlated for the purpose of NetOps.

6.1.1 Pooling Data and Accessibility to Information

A single Splunk database can be built distributed and paralleled, so that increased size does not imply retrieval delay increase [27]. Splunk's interface APIs allows for easy

access of other systems to utilize the information indexed across multiple organizations, by giving a capability to independently deploy and parallelize its different components to their own dedicated hosts such as the deployment server, indexers, and search heads [28]. The Navy could potentially build one major Splunk instance and create policy requiring the synchronization of information across every Navy organization/command. In theory this would then pool all of the Navy's network information into a single indexed database, controlled by a Splunk interface. This type of framework would be cost-efficient if managed under the same Splunk cost account, as data indexing prices taper off as more information gets indexed as explained in Section 5.2.2. Also assuming that the small use-case at NPS scales up, this would naturally employ the use of the DevOps in the process.

This simple solution would be cost effective, and increase the effectiveness of NetOps in Navy enterprise operations significantly by providing easy accessibility to information across all networks. NNWC and NCDOC would then need to only focus on Splunk deployments and the quality of data being pulled up to a single accessible Splunk instance at the highest level. Splunk event correlations, machine learning ability, and other tools could be the building-blocks of automated actions. Any command could then develop new automated solutions based off data correlation alerts. Automation, in general, would then be a natural next step after this data-pooling solution has been partially matured, and DevOps automation actions would likely follow. Over time, the quality of the information, in theory, would get better, allowing for granularity in the automation.

6.1.2 Weakness in Adoption of Splunk as A Full Solution

While Splunk may seem like a good solution since the company is dependable and its software is best in class, a full-blown adoption of its software by the Navy threatens the infrastructure on the reliability of an external organization. Not that Splunk is not reliable, but a military organization should abstract away third-party architecture dependencies.

If Splunk one-day went bankrupt, its software would cease to be supported. This would mean the architectures built around the Splunk solution, would need to be changed. Change in a networking architecture is very expensive because it would mean re-wiring the informational flow to multiple components which once relied upon input from the Splunk interface.

Another reason why Navy-wide adoption of Splunk is at best a short-term solution is the

continuous need for unique use-case adaptations to a specific adoption of Splunk into a specific architecture.

In conclusion, the employment of Splunk at the service-level, is still at best a good short-term solution for Navy commands in data consolidation. A longer-term solution could, however, incorporate the functionality of Splunk while also abstracting away the necessity of dependent architecture. By developing a Navy Open Architecture and interface to support DevOps, this would allow for compatibility with Splunk and other tools.

6.1.3 No such thing as a Simple Framework for Automation of NENs

The practice of DevOps requires that there be specific tools on top of any architecture that allows direct access to low-level system manipulation (development), testing, and rapid-deployment. Splunk does not provide any such capability, nor has there been any common simple solution to this problem seen in this research. AT&T in Section 5.1.1 built its own automation software from scratch and found that a shift to virtualization was key to future automation deployments. They prompted development of their own open software suite ONAP for just that purpose. Extreme provides automation capabilities in its offered software suite, outlined in Section 5.1.2, but the entire premise of the utilization is that it employs its software across the entire enterprise, which would mean a complex technological adoption across every network in the enterprise. For the Navy, that would also mean tossing out old solutions built with specific automative purposes in mind and needing to re-design solutions with the new adopted software.

Both AT&T and Extreme's solutions would need a cost analysis, and numerous benefit trade-offs that must be compared and contrasted prior to picking one or the other. Also, there are numerous other competing automation technology platforms not specifically mentioned in this research but are just as relevant, such as IBM's Netcool, BMC's Helix or even RedHat's Ansible tool suite.

To further complicate this problem, tools most often are mixed by different organizations, which blend a mix of best practices. One such Navy POR, which has already done this, is PMW 160 in the development of the CANES/ACS system, which could be considered its own tool suite.

ACS documentation and binaries with supporting IAC scripts are hosted on a publicly available site called Defense Intelligence Information Enterprise (DI2E-DEVTOOLS). DI2E-DEVTOOLS has a publicly available ATO for this system. According to the hosting site:

> The DI2E Developer Collaboration Tools provide documentation and design artifact hosting, issue tracking, and project collaboration. The tools enable automating nightly builds and tests of software projects flagging and sending errors to development team(s) for action, automating unit and IV&V tests sending results to development team(s) upon completion, check-in/check-out capability to maintain versioning of software baselines in shareable source code and design artifact repositories. The DI2E Developer Collaboration Tools provide an environment where code produced for the government is configuration managed and buildable on government owned property. [29]

Since it is a publicly available ATO, this means any government organization can mimic the capability for themselves, and adopt the technology. The ACS environment offers the framework needed to manage the automation of complex networking systems, and it is already employed by the Navy's afloat network. This software suite is very important because it is built with idea of system integration in mind, e.g., to enable multi-platform (ship) systems to communicate and develop new applications. ACS also was specifically picked to be part of the C2C24, which is further discussed in Section 4.5. Figure 6.1 illustrates how ACS might deploy an application.

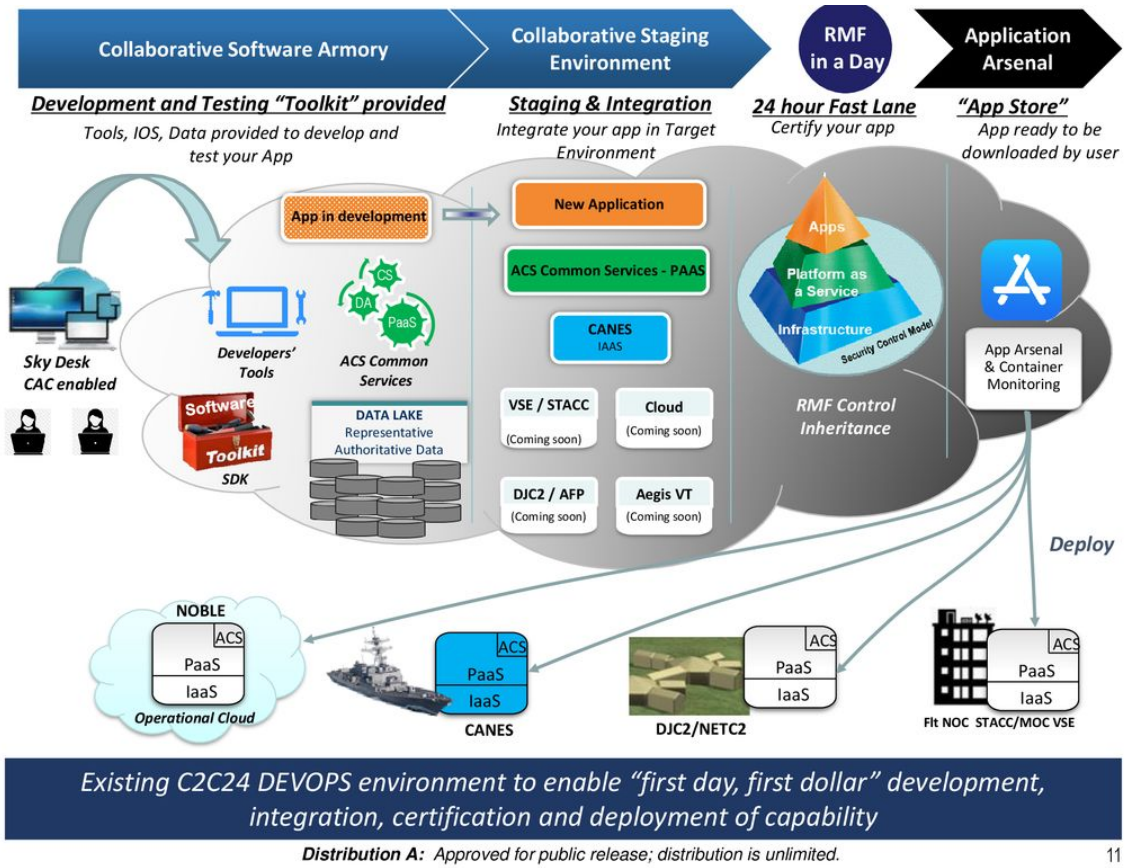


Figure 6.1. ACS's Role in C2C24. Source: [30]

Picking automation technologies, or picking an automation platform requires an on-going continued effort for the Navy. There is no simple solution to this problem, and there are multiple PORs unmentioned in this research that deploy their own separate automation frameworks, further complicating the matter. Adoption of the ACS suite of tools by all NEN, as the Navy's universal framework in automation may, however, be a potential solution to employment of an automation framework. It is the first in adoption of C2C24 and this seems to be well aligned with the goals of DevOps. CHIPS Magazine quoted ADM Barrett saying the following in regards to C2C24:

> If I construct applications in a different way, where I use the shared infrastructure of the ship, I can focus on the capability delivery and not on providing infrastructure which takes a long time," she said. "Then I can inherit all of

> the Risk Management Framework cybersecurity controls and all of the accreditation that's already been done on that shared infrastructure provided to me. I can 'drop code, not boxes.' And if I use standard web ports and protocols, standardize my data, and adhere to the development environment standards like commercial industry does, I can automate the RMF process for the micro service, because it becomes just testing that containerized piece of additional code — you don't have go back and retest a highly integrated application that may not work in the environment. [31]

The goal for C2C24 seems to align well with the goals for DevOps, so utilizing ACS as a potential common platform for automation framework of NEN processes is an avenue for future research. Funding for the project outlined in Chapter 2, has been allocated into an Amazon Web Services (AWS) Government Cloud hosted by NPS to test the hypothesis that all NEN functions can be automated with ACS by exploring automation of the use-cases outlined in this thesis.

6.2 A Long-term Solution in Automation of Navy Networks

The short-term solution presented in Section 6.1.1 is incomplete because in reality automation, similar to network security should be baked into the employed network architecture. If the best commercial automation tools were to offer any hint as to the secret of good automation or "automation best-practices", the one-thing in common between them seems to be the need to have a shared automation framework for all networking tools, enabling DevOps to best thrive in the organization. This also confirms the statements made in Section 3.2 regarding the concept of DevOps, and how DevOps thrives when specific kinds of tools/capabilities are made readily available to teams. Assuming there was commonality across all of the tools that enable organizations to do DevOps, they could be thought of as an "automation framework", hence the use of the term.

There are multiple problems with deploying an automation framework, and the majority of them are not technical. The main inhibitors to progress in the Navy are generally organizational issues such as funding, bureaucracy, special interests, acquisition life-cycles, and bad policy. All of these topics are suggested as avenues of ongoing research and

potential future research topics in the conclusion of this thesis; however, since those topics generally affect NetOps, there is a framework called INOSS getting worked on by FCC to try and get around those issues and a multitude of others. The recommendation for a long-term employment of automation is through use of INOSS, which, in theory, would make this solution a part of a larger organizational framework that will hopefully abstract away some of the larger organizational problems, specifically the automation problems outlined in Section 3.2.

6.2.1 Integrated Navy Operations Support System

FCC's current Executive Director Mr. Manuel N. Hermosilla was employed just after his role at DISA as the Chief Engineer for Network Management directing DISA in its Global Information Grid Bandwidth Expansion Program. During this project, his team built and deployed DISA's current framework similar to Integrated Navy Operations Support System (INOSS), exacting the changes needed to unify the organization's then patchwork architecture to include the necessary changes in the organization's budget, acquisition, and policy to streamline the technical framework. INOSS is the newest project, which, should exact the same changes in Navy networks as Mr. Hermosilla did for DISA.

The fundamental ideas behind the INOSS framework according to its Concept of Operations (CONOPS) are as follows:

> The operational framework must be refined and evolved over time to adapt to changing capabilities and mission needs. The Navy Cyberspace Operations Architect (NCOA) ensures the FLTCYBERCOM/TENTHFLT operating model defines common, global, cross-program operational requirements for DoDIN-N operations and Defensive Cyberspace Operations (DCO). The NCOA will work with operational stakeholder organizations to
>
> - Adopt and consistently promulgate a common, standards-based reference framework for Department of Defense Information Network - Navy (DODIN-N) operations and Defensive Cyberspace Operations (DCO)
> - Ensure that the diverse development efforts to meet operational needs support global DODIN-N ecosystem objectives, are mutually consistent and interoperable, and conform to a strategic vision

- Ensure the Cyberspace operational requirements needed to proactively manage the DODIN-N ecosystem are communicated to the Office of the OPNAV, U.S. Naval Information Forces (NAVIFOR), and Space and Naval Warfare Systems Command (SPAWAR)

Comprehensive SA and effective network maneuver rest on a foundation of core capabilities that include:

- Architecture: a planned, unified, strategic approach for operating, managing, and defending the DODIN-N
- Asset and Configuration Management (CM): an accurate, current, complete inventory of DODIN-N network resources and their configuration. [1]

The CONOPS covers many of the important political topics such as the scope of INOSS, operational relationships, task force alignment, unified global C2, and operational tenets. This is important because it is a holistic framework where technical network architectural issues such as information sharing, configuration management, and access controls can be addressed without worry of interfering organizational issues.

INOSS spans across every POR at every classification level and is built to include not only all NENs but also excepted networks such as NPS. The best way to describe it is to say it is a "technology-neutral architecture", that facilitates the interactions of technology. A picture from the CONOPS to describe it at the highest level is found in Figure 6.2 depicting how the "INOSS-stacks", otherwise referred to as a frameworks, are built at every classification level to roll up information to the PORs for adoption of universally applied operational tenants to include SA, Configuration Management, C2, and Risk Management.

Figure 6.2. Navy's Operational Environment Aligned to the INOSS Framework. Source: [1]

6.2.2 The INOSS Framework

The FCC proposed INOSS framework is best depicted by Figure 6.3. This figure describes how data is first acquired at the lowest tier labeled as "DODIN-RESOURCES." It then is moved up and through the tiers eventually transitioning data to information and information into knowledge at the top level where C2, SA and general decisions can be made.

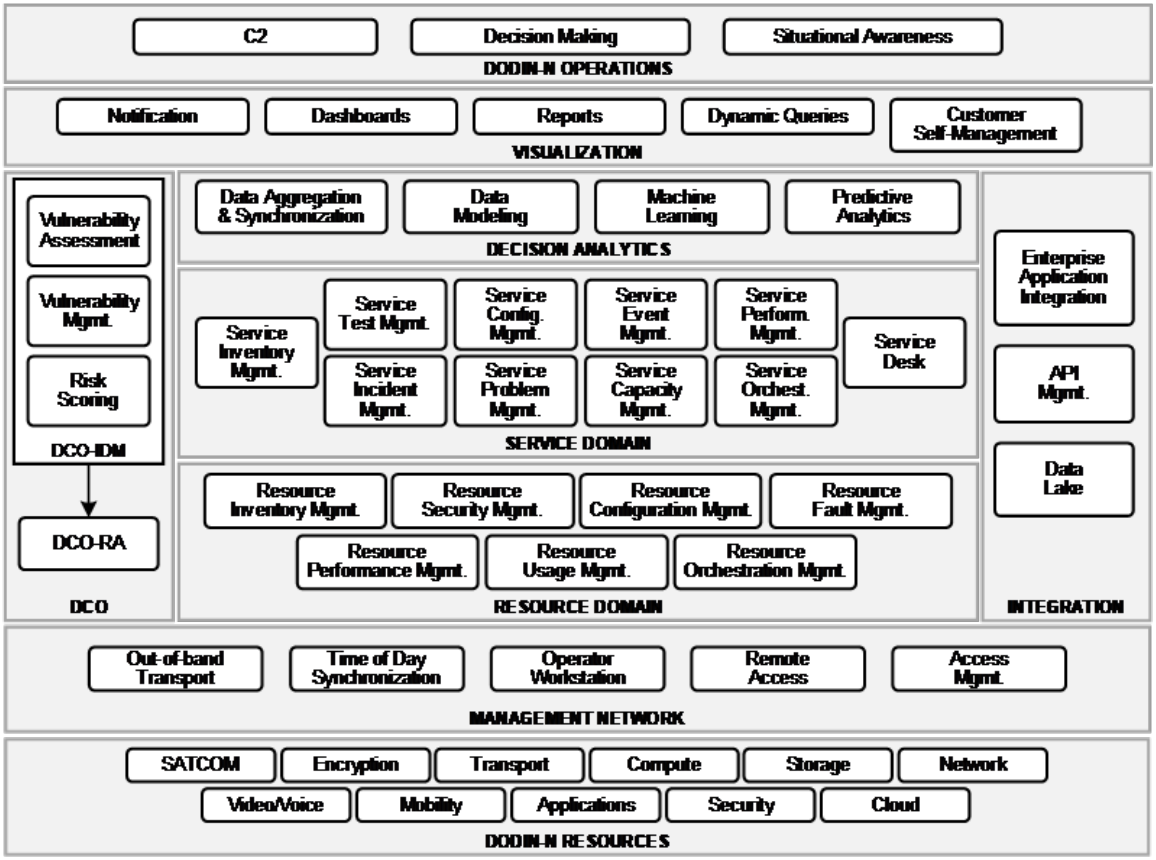


Figure 6.3. FCC's proposed INOSS Framework. Source: [1]

What makes this a framework abstracted away from technology is the fact that once adopted, the defined services of this framework utilize any capability to perform a semantically defined function. Every block in the framework will have integration rules that it must follow. In order for a technological solution to be implemented, the technological solution passes information through a defined interface. Figure 6.4 depicts graphically how this would work, where technology is decoupled from function, which the INOSS framework enforces. Then, if technology changes, the framework can simply swap it out without affecting the integration of other tiers or internal tier components.

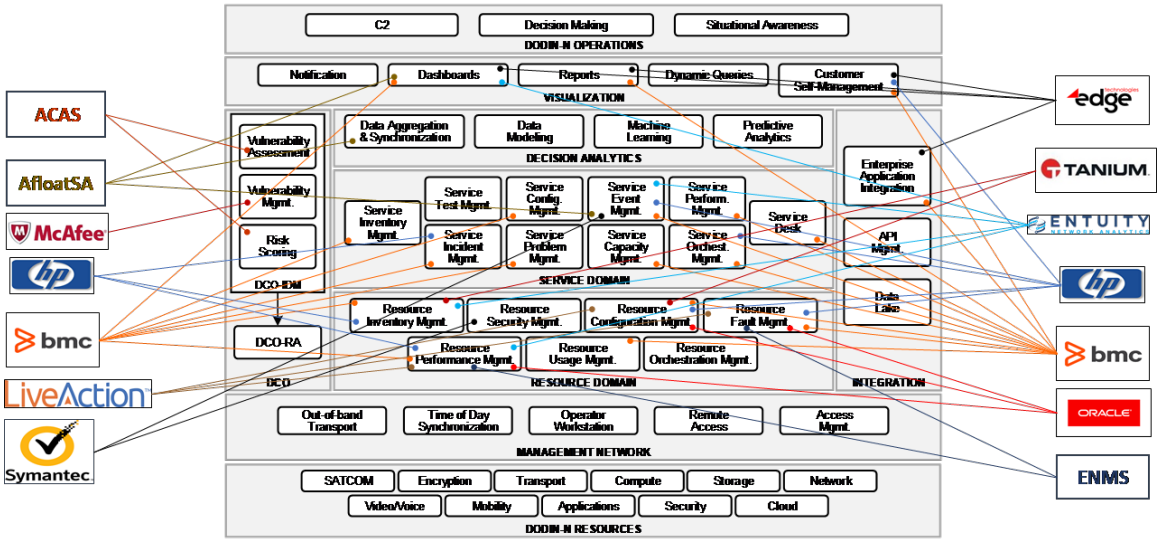


Figure 6.4. Mapping of Representative Tools to the INOSS Framework. Source: [32]

Figure 6.4 represents the mapping of current technologies pulled out of context from the *INOSS Gap Analysis*, which is looking at tools employed "currently" not necessarily "correctly". Most have overlapping functions. The gap analysis specifically says:

> In many places, there is overlap in either functional capability or managed resource coverage. It is important to note that these functions are defined at a very high level, as they are intended at this stage to provide a quick view of any obvious areas of support which may need additional attention, and specific management capabilities may vary for any given system based on scope and mission objectives of the system. [32]

Overlapping of functions is not the ideal use of INOSS, but it portrays the point that streamlining functionalities is the goal of INOSS. So, a lovely added benefit of this framework is that it allows for an analytical view of all the tools that may in turn be re-used/added by another POR, with the goal to ensure a common adoption of technologies/capabilities.

6.2.3 Data-Consolidation Covered By INOSS

INOSS covers the first need to have a unified data consolidation framework by implementation of the "Integration Tier". INOSS would define all of the fine-grained functional implementations necessary and it becomes a layer of abstraction from the network architecture to absolve away all third-party implementation issues explored in Section 6.1.1 where Splunk provides the API management and in the case of PMW-160 in Section 6.1.2 the "data-lake". Figure 6.4 shows the data-lake is in fact a GAP in the architecture which INOSS would need to resolve. INOSS could easily map its data-lake to the tool being developed at NNWC explained in Section 5.2.4. As a result, if INOSS were put in place, all PORs would have that data-lake as "the" mapped solution for data-consolidation. In addition, because the "API Management" is part of INOSS, then all application interaction and associated tools could follow a universally defined INOSS interface (or supported tool) to consolidated information. Thus, if there was a Splunk interface to the data-lake, that interface would first just be mapped to the INOSS defined API through the "API Mgmt" interface so that any future technology after Splunk would only need to cater to the rules defined by INOSS and all internal integration of the network remain decoupled away from Splunk.

6.3 Suggested Changes in INOSS

The best functionality found to mimic the need for DevOps in INOSS is defined at the "Service Orchestration Mgmt" and "Resource Orchestration Mgmt" modules. According to the *INOSS Framework* the descriptions of those two orchestration modules are:

> Service Orchestration Management applications and processes perform automated service decomposition, service integration, coordination, and management of service deployment, service activation, and service assurance. Service Orchestration Management is directly involved with service fulfillment activities for delivering Cyberspace capabilities, and the automated remediation tasks for service assurance and security. [8]

and resource orchestration is: "Resource Usage Management involves the applications and processes responsible for the collection and reporting of resource usage to higher order OSS tools and processes in the Service Domain, such as Service Capacity Management" [8].

The job of the "Resource Orchestration Mgmt" module is straight forward as far as collecting usage of the resource domain to report up to the service domain. This job will be done by tools/processes that gather information only at the resource tier, which does not impact any other tier, and essentially feeds the information up to the Service Domain. It seems; however, some futuristic capabilities and automation issues are overlooked by placing "Service Orchestration Mgmt" within the Service Domain.

DevOps needs to be able to affect automated processes at an inter-service level. If DevOps is a part of the process, then this service will be limited to only the "Service Domain" tier where it resides. If the job of "Service Orchestration Mgmt" module is simply to perform automated service decomposition and integration, it cannot perform those actions from the service domain tier. There currently is not a good mapping of DevOps type tools for automation in this architecture.

A change to the INOSS framework that would allow for an "automation framework" is necessary in order to ensure that future automation frameworks are abstracted from their actual tool-sets and enacted via a de-coupled process.

6.3.1 Breaking Down the Changes

There are several proposed changes in red on Figure 6.5. This section will describe each of those changes. First there is the added new tier called "Service Orchestration and Automation" with its four new modules. Next there is the extension of the "Integration" tier to reach both the "Visualization" and "DODIN-N Operations" tiers and the addition of a new module called "Automation Integration". Another change is the addition of "Governance" to the Service Domain Tier, and lastly "Data Aggregation & Synchronization" has a recommended change in its verbiage.

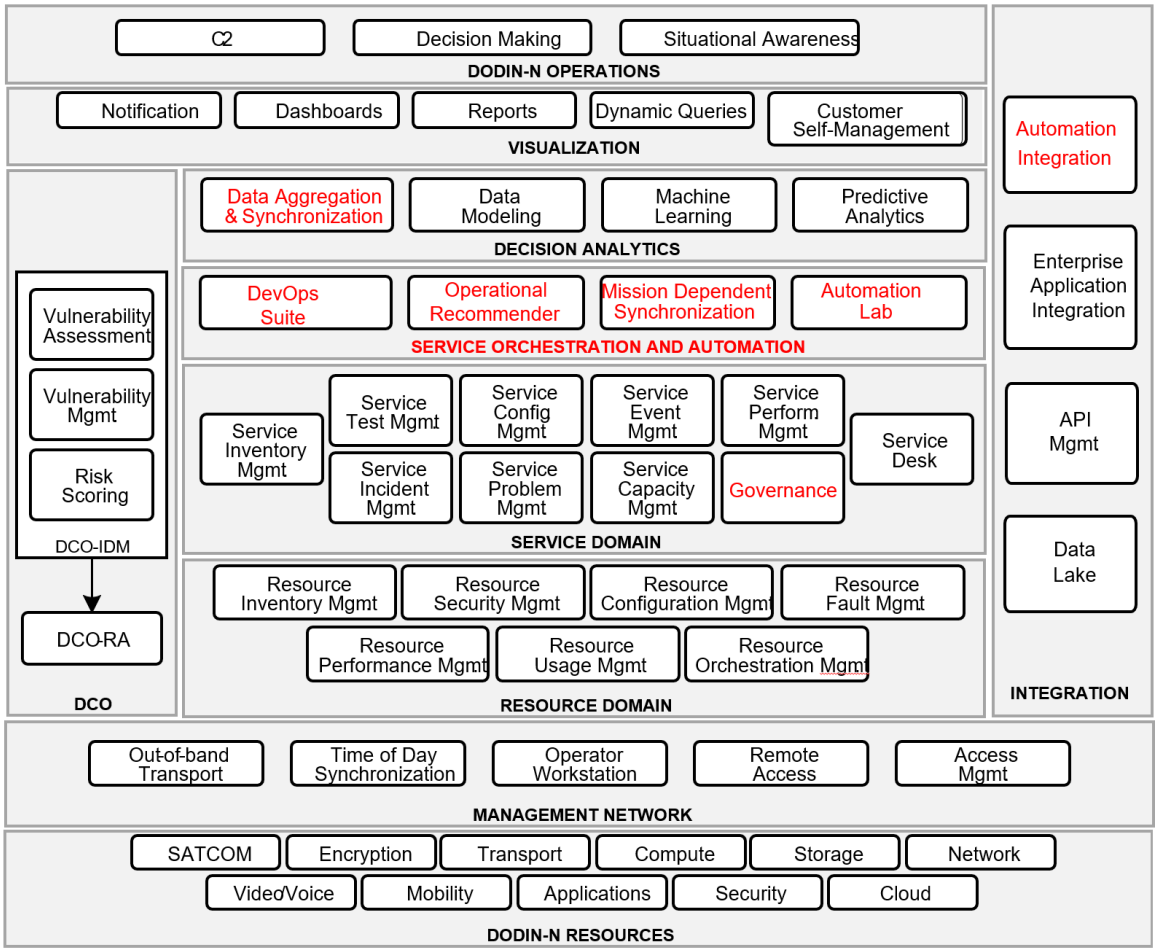


Figure 6.5. A Proposed Change to the INOSS Framework, (changes in red). Adapted From: [1]

Service Orchestration and Automation Tier

The first suggested module titled "DevOps Suite" represents the tools and processes needed by a DevOps team that work to identify and remove repetitive tasks identified in all services. As we all know, there are a multitude of service-level jobs that need no human to accomplish, as much of the work can be repetitive. This module of the tier would interface DevOps tools and ensure automated task/processes are constantly being evaluated. Process management in automations should remain trimmed else they become unwieldy, so this module in effect offers a location to place all tools that perform any additional service to remove redundant operational tasks. This would benefit any future GAP analysis, as multiple redundant

DevOps tools exist, and give insight when redundant DevOps processes have been put in place.

The second suggested module "Operational Recommender" provides services used to identify places where decision analytics need new automated processes in place. A recommender service is used in automation to deliver new machine-learned processes. This module already has technological tools developed to perform these kinds of tasks in industry, and they will be an important part of future Artificial Intelligence (AI) integration for getting knowledge to the visualization tier. The Recommender module would likely integrate with capabilities maintained at the Decision analytics tier. Thus, its placement here rolls up automated recommended information for potential knowledge extraction in analytics.

The third suggested module "Mission Dependent Synchronization" is a future technology anticipated as a necessity in military operational NetOps. In cases where applications are built around latest "mission-mapping" efforts of the Navy, automated synchronizations of service mission can utilize this module. Automation's utilizing IAC processes, such as Terraform Scripts are envisioned to live here. These are automations that would enable changes to network infrastructure, or inter-service relationships. Here the availability, redundancy, or COOP efforts are based on constantly changing real-time operational maneuvers.

The fourth and final suggested module, "Automation Lab", is straight forward. This is where all automation testing tools/processes could be performed. The automation lab provides environmental prototypes to test within and would be the module where inter-service orchestration and automation integrations would also be tested. Futuristic capabilities such as PMW-160's "Digital Twin" tool simulation of a shipboard environment would be mapped to this module.

Integration Tier

The INOSS integration tier ties in the modules that worked across the "Decision Analytics", "Service Domain", "Resource Domain" and "DCO" tiers. Changes here suggest the extension of this tier into the "Visualization" and "DODIN-N Operations tiers. It also proposes adding an additional module "Automation Integration".

Extension of the Integration Tier

The reasoning for the extension of the integration block into the Visualization and DODIN-N Operations is due to the fact that changes in the integration can affect overall visualization and operations. A change to an API can easily affect change in visual information, or at least give potential for change in the visual information. Also, the addition of automation integration may be able to interface with visualization or C2 capabilities. Not having this change, illustrates there is no human out of the loop for both tiers, which inhibits DevOps capabilities if that is the intention.

Adding Automation Integration

The "Automation Integration" module has been added to show capabilities for automation actions for integration into all levels of the framework. Automation Integration might be in the form of the API Management or Changes to Enterprise Applications; however, this would not cover potential interfacing technology integration between INOSS tiers and modules. Technology that automates "INOSS" interfacing technologies would live here.

Governance Module

The "Governance" module has been added to specifically incorporate all network governance capabilities into a single Service Domain module. Most commercial industry maintains multiple "governance" tools used to enforce policy and to audit the network. If the network violates an SLA or rules as defined by governance, the result is often penalties that must be paid and of course actions that must be taken (which can be automated).

Changes to Data Aggregation Synchronization

The "Data Aggregation and Synchronization" module highlighted in red in the Decision Analytics tier. The description of this function seems to be missing a key component which is vital to success of the overall Decision Analytics block.

According to the *INOSS Framework* [8] the Decision Analytics tier will "supplement traditional Operations Support System (OSS) tools with the following: uncover insights from data not currently processed by the OSS tools". Even though this is mentioned, there is no follow up to this type of action to be covered in its modules. The module that seems as though it should support this type of behavior would be the "Data Aggregation and

Synchronization" function. The description of this section of the *INOSS Framework* is as follows:

> Data aggregation and synchronization is essential for applying analytics. In order to apply data science principles, techniques, and algorithms to disparate input sources, it is necessary to homogenize the data.
>
> Information must be aggregated and correlated to provide a more robust understanding of activity beyond the current level of understanding conveyed within any single data-set. Within the construct of the Cybersecurity Reference Architecture (CSRA), the Navy NetOps and Cyber Security Centers will develop a construct for sharing information and analytics for development of improved data analytics across the coordinated functions of maintain, operate, and defend.
>
> Data aggregation and synchronization support the functions and processes involved with the acquisition and ingestion of data from multiple concurrent and federated sources. This includes capabilities that ensure guaranteed delivery of data and integration with D-DIL environments.

In addition to these over-arching specifications, recommend that INOSS also incorporate the additional specification:

Ensures adequate data acquisitions are levied as much as possible. Where there is an informational gap in hardware analysis, due to hardware or software limitations of equipment, tools are identified, evaluated, categorized and put into context of the Open Systems Interconnection (OSI) model layer from which its data originates.

One way this might be implemented is by evaluation of tools by their capability to provide raw data at every level, to include physical level data such as power consumption, or even power jitters. Industry considers this to be the very first step in doing any data analysis. Without fine-grained adequate reliable data, industry can lose a valuable competitive edge. Analysis will be less effective if all data is provided only by software capabilities built to "infer" from informational gaps, which is not as good as hardware with raw sensor coverage. A hard lesson learned by Extreme Networking from a prior Uber employee found vendor software had multiple hardware limitations keeping them from fully understanding

a problem until updated equipment was added, or software was changed to ensure data was properly acquired. [21]

6.3.2 Interconnection of INOSS Tiers and Modules

Section 6.3 explained some of the additions to the concepts of the INOSS framework to support automation; however, it did not mention how INOSS will implement the ability to "absolve away all third-party implementation issues". The INOSS framework, which is still in its infancy, needs the ability to de-couple operational functions across its framework. As of now, the most useful function INOSS brings is the ability to do a functionality GAP analysis over a networked system.

The suggested tool for dealing with the interconnection inside INOSS is the Army developed capability called Future Airborne Capability Environment (FACE). Originally funded to be an interfacing language for embedded systems in specifically air craft systems, FACE has evolved to encompass much more. Its objectives are spelled out in the *FACE Technical Standard, Edition 3.0* which states most "software systems today are tightly-coupled integrations of software components without regard to portability" [33]. The idea behind FACE is to allow for software to remain flexible and avoid what FACE refers to as "vendor lock", meaning a third party company software is not the only software around that will allow the technology to function.

Mostly FACE applies the following software engineering practices which would benefit INOSS if applied to the framework:

- Use published industry standards to provide normative references, allowing the use of existing software libraries and tools whenever possible
- Use profiles to define subsets of those standards when support of the entire standard would lead to safety or security certification issues, or when supporting only a defined subset would lead to a more cost-effective solution; a profile can also reference a specific version of a standard in its entirety
- Use a standardized architecture describing a conceptual breakdown of functionality and the FACE Reference Architecture to promote the reuse

of software components to share common functionality across military systems

- Define standardized interfaces to allow software components to be moved between systems developed by different vendors
- Use a data architecture to ensure the data communicated between the software components is fully described to facilitate the integration on new systems
- Require that hardware abstraction be used to decouple software components from specific hardware implementations, and device driver normalization be used to allow interfaces to external devices to be developed independently of the computing platform device drivers
- Use a display window management strategy to incorporate common avionics user interface standards to aid in the integration of components needing to share display areas and input devices. [33]

The idea here is that FACE could potentially be the interconnecting language between components in INOSS, allowing INOSS to solidify its current framework and decouple capability interconnections. This would also mean any system such as ACS/CANES or even ENMS, which currently do not utilize FACE must implement the standard. As Army and Air Force's adoption of FACE continues with systems such as the F-35 or R2C2, a military radio system software component that is easily able to be integrated into any FACE compliant aviation platform [34], even more adoption of the FACE standard will continue to get built. If INOSS adopts the FACE standard, it has the potential to open up the Navy's networking architecture to utilize and support joint service system capabilities. Also, NPS now has the ability to test the FACE integration on ACS because FACE is an openly available standard. Future research into integration of the INOSS framework will be suggested in the conclusion of this thesis.

6.4 INOSS Integration Does not Mean DevOps

If INOSS with automation suggestions were enforced by the Navy, this would, in turn, force Navy PORs to adopt a common automation framework. Having that common automation

framework could be a driving factor for the cultural shift to DevOps employment and some of the other issues discussed in Section 3.2.1 about the barriers in DevOps. However, even if DevOps *can* be fully adopted, that does not necessarily mean that DevOps *will* be fully adopted.

DevOps must first be embraced by changing the way operations are conducted, and specifically adopting policy that enables DevOps to coexist alongside network operators. Specific to the mission of NetOps across NENs a likely location for DevOps team members, is in doing micro-service management and being physically located at every NOC, NNWC and NCDOC where operational NetOps automation can be most clearly defined. DevOps team members need to work closely with the NetOps teams assisting in every day redundant processes to better observe which behaviors need automated.

6.4.1 Onsite DevOps in NENs

Section 3.2 does a breakdown of the DOD's publication about DEVSECOPS titled *DOD Enterprise DevSecOps Reference Design*. The document mentions collaboration tools that help facilitate communication in DevOps, but nowhere does it mention the physical proximity of Operators to Developers. This section explains why the most beneficial communication practice is to keep developers in close physical proximity to operators.

If MUOS Satellite Logs (discussed in Section 4.4.4) upload log data to a shared data-lake capability, DevOps teams would be able to automate data correlation analysis. When a crisis is unfolding, the log information of MUOS satellites could be a common resource across the NENs for troubleshooting. A DevOps team could enable that information in automate MUOS troubleshooting procedures. Other DevOps enhancements such as auto-COMSPOT creation when connections go down unexpectedly and the ability for a NOC to update and share real-time MUOS status with tools such as NCSA would be feasible and potentially developed quickly. Additional correlation capabilities such as the ability to use deep learning to anticipate when MUOS outages will occur, would give operators a head-start on actions to be taken before events occur.

With DevOps comes the risk of deploying system degrading code. The development of these operational enhancements needs to be balanced with the understanding that operators use/access all of the tools effected, and new deployments need to be actively monitored

onsite by a DevOps team member just as an operator is tasked to be on watch 24/7. DevOps team members onsite would have the ability to roll-back changes if an adverse reaction changed the production environment. There may also be further need to automate even more actions depending on how operators generally respond to events. Having an onsite DevOps team can use operator knowledge and automatically employ operators to test solutions on-site, prior to deployment. For example the redundancy of COMSPOT messages with ENMS ticket creation over-viewed in Section 4.4.3 could be addressed easily if an ENMS DevOps team member on any NOC watch-floor had the ability to observe the ticketing procedures which operators are employing, and automate auto COMSPOT creation. Having a ENMS DevOps team member on staff at every NOC would enable that team to be able to synchronize efforts through their communication at every point in the network, just as suggested in *DevOps A Software Architect's Perspective* and mentioned in Section 3.2 in the suggestion that effective employment of DevOps is through the use of small teams to coordinate communications.

The PLA system with its synchronization and inefficient communication of new additions, outlined in Section 4.4.2 would clearly benefit from enhanced automations that a DevOps team member would be able to employ. Where the OPNAV SNDL needs to have a more efficient method in updating its base command PLAs, the team of operators at NCTAMSs also could streamline actions in keeping its own PLA list accurate and synchronized across both NCTAMS.

These example use-cases outlined in this thesis only serve to show a small fraction of the potential interactions a DevOps team could find and solve by having an onsite (NOC level) presence. The conjecture is that the first level of NetOps operational information roll-up is happening at the NOC. As this is a natural launch point for ensuring quality, integrity and availability of systems and their data, there is near unlimited work to be done to streamline processing the information through automation practices. Also, there is the assumption that more of these use-cases as well as the detailed minutiae will be found by smart and actively seeking onsite DevOps team members. Part of a future study will need to show that PORs cannot employ real DevOps unless there is a persistent onsite level of engagement in the NetOps mission at the NOC level.

6.4.2 How do NENs Shift to DevOps Employment?

This study has shown potential ways for NENs to adopt automation tools into a framework that could potentially benefit DevOps. It has also described why automation enabled NENs can still potentially fail to employ DevOps even with all of the correct tools. Policy, culture, and organizational alignment to conducting NetOps filtered by DevOps needs to be adopted by the Navy in order to enable DevOps friendly procedures.

The first step to solving this problem is to ensure that the automation framework is in-place. The next steps will be dependent on the capabilities given in the first step, but ultimately, they must align with the Navy's willingness to allow risk for potential network degradations because even though DevOps procedures are fast, they come with an added measure of risk for bugs in the deployed solutions. Also, anticipate that there will be an up-front cost in changing NENs processes. Chapter 7 outlines a few areas of future study where policy, culture and organizational alignment are suggested.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 7:
Final Conclusion and Future Areas of Study

Based on qualitative research, this thesis concludes that current NEN automation of NetOps functions will not enable the Navy to pro-actively operate its network without additional changes. It also confirms at the end of Chapter 5, two necessary changes based on best-practices to enable future pro-activeness in NetOps:

1. Provide a Navy-wide interfaced information repository
2. Incorporate a Navy-wide DevOps automation framework

Chapter 6 proposes a short-term and a long-term solution to incorporate the proposed changes mentioned above.

The short-term solution suggests using Splunk as an efficient single tool to enable commands to crawl networks and correlate information. This recommendation is short term due to the fact there are many disadvantages in the direct use of Splunk as a Navy-wide data solution due to third-party dependencies and the need for unique use-case adaptations for each system where it is employed. Splunk, however, could potentially satisfy the first suggested change to Navy-wide use of an interfaced information repository. If employed by the NEN, at the very least, it would potentially be able to identify the gaps in its automation deployment as information gaps become apparent.

The long-term solution recommended by this thesis employs the use of FCC's INOSS framework. INOSS with a few suggested changes has the potential to bring Navy networks into the modern era of DevOps automation practices. INOSS also has the ability to potentially take the recommended short-term solution and abstract away the third-party dependency issue, enabling commands to adopt Splunk technologies with eventual transition over to INOSS supported technologies when ready.

7.1 Future Areas of Study

7.1.1 FACE integration with ACS

Section 6.3.2 explains the need for interconnecting the INOSS framework by use of interfacing components and tiers. FACE was recommended as the potential solution to this type of problem. There is potential here for work taking NPS's ACS prototype and utilizing FACE to connect to another FACE product. The value of showing that a large enterprise application like ACS is able to adopt a FACE interface would imply that the INOSS framework should be able to define its modules in a similar fashion.

7.1.2 FACE integration with INOSS

FCC currently supports the initiative to organize and consolidate capabilities into the INOSS framework. Research to investigate how INOSS can be shaped to support FACE integration is recommended. There is potential for INOSS to support enterprise IT systems as well as smaller platform IT solutions. Having capabilities mapped to support the FACE interfacing semantics would open up INOSS to allow for any component developed with FACE integration both internal and external to the Navy, to interact with joint and commercial systems already developed to comply with FACE.

Doing this work upfront would ensure code redeployment is streamlined, and the need for future money spent on architecture and design (one of the highest costs for re-design) is limited. FACE integration also means that a built-in system for software pedigree would be included, enabling built-in safety standards and security of artifacts.

7.1.3 PORs Cannot Employ DevOps Automation Without an Onsite Presence

Section 6.4.1 gives three good examples why an onsite DevOps is needed for finding and fixing of automation issues in a NEN. A study on the direct impact of onsite deployment of DevOps at all NOCs, NNWC, and NCDOC and whether or not it would be beneficial to the development of automation in operational information consolidation and C2 of the Network. This type of study combined with a cost analysis of staffing onsite DevOps teams would provide a solid understanding of this need to staff these organizations with adequate

personnel for the job, and possibly provide leverage to operational commanders to task PORs to fulfill these positions.

7.1.4 Automation at the Unit Level vice the NOC Level

Using NPS's ACS prototype, one could potentially test automation deployment practices from unit level networking instances. Proof that unit level automation deployments can be done effectively and synchronized across other units, would be very valuable point to support the use of local network centric data-analysts in a Navy strike group. Showing this type of capability could be done by enabling certain ACS instances with DevOps type capabilities already offered in the ACS tools suite.

7.1.5 Enforcement of NetOps

In Section 4.4.2 a problem is mentioned regarding C2 of NetOps actions through the Navy's ADCON. Further research of this topic will reveal that a Navy POR called VRAM is tool used to enforce operational orders execution. NetOps is unique in that operational commanders have an interest in the execution of NetOps but the Navy must leverage its ADCON as the enforcement mechanism for non-compliance. OPNAV is the organization in the Navy which maintains its ADCON relationships of all commands and publishes them to a document called the SNDL. The PLA of a command is a direct linkage to the profile for a command in VRAM. The way OPNAV updates the SNDL is by having a command register its PLA via some internal OPNAV process. The way a PLA is assigned and updated described in Section 4.4.2, is another completely separate process from the one used by OPNAV. Where two separate methods of PLA updates are maintained via human processes, the two lists are never compared, and rarely match, causing issues in the enforcement of NetOps. Streamlining this capability, by automation of PLA synchronization between OPNAV and NCTAMS could be a way to assist with this issue of enforcement.

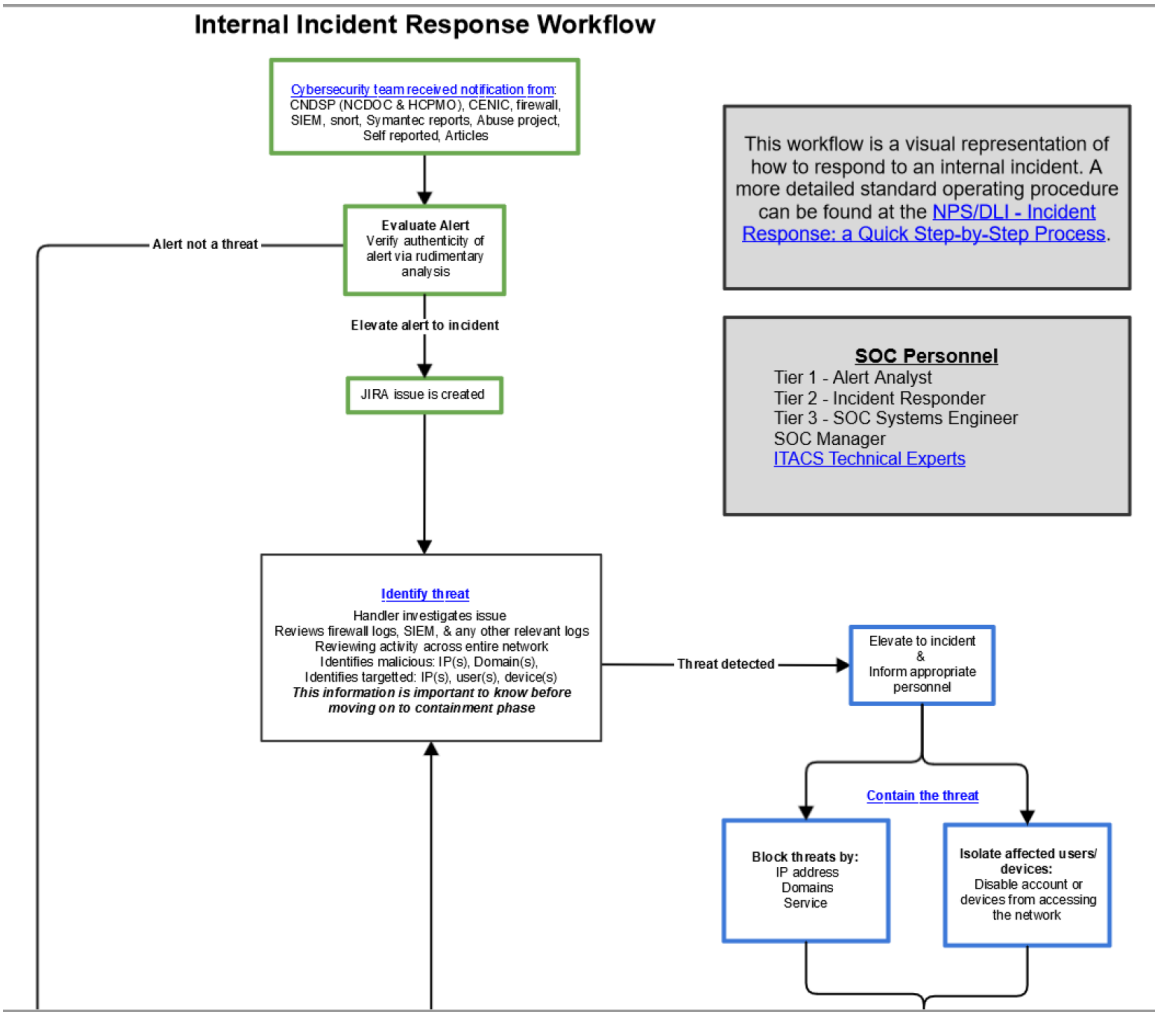
7.1.6 Changes to the DoD Acquisition Lifecycle

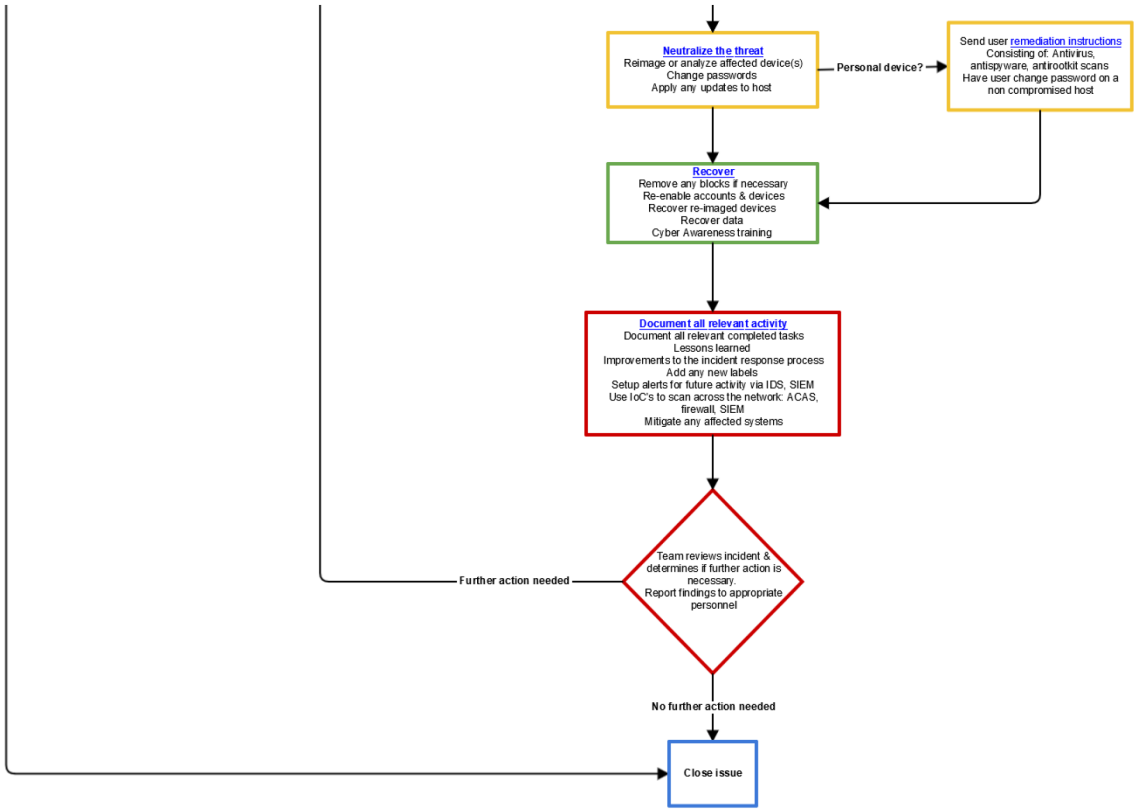
Barriers to DevOps were noted in Section 3.2.1 to include the culture of operations, cost and displacement of personnel, and changes to policy. The Defense Innovation Board (DIB) draft report titled *Software is Never Done: Refactoring the Acquisition Code for Competitive Advantage*, explains findings for why key changes are necessary for the DoD to enable

DevOps [10]. This research could be leveraged in addition to the findings from the DIB, to begin a study on effectively employed DoD instances of DEVSECOPS which were recently formalized in April when the DoD published the article: *DOD Enterprise DevSecOps Reference Design* [9]. Finding where the DoD has already successfully employed formalized instances of DEVSECOPS could lead to use-cases that directly assist with the development of policy for Navy organizations to use in the adoption of a shared operational framework such as INOSS

APPENDIX A:
NPS Incident Response Work-flow

NPS documentation of routine incident response automation. Used for DevOps process team training.

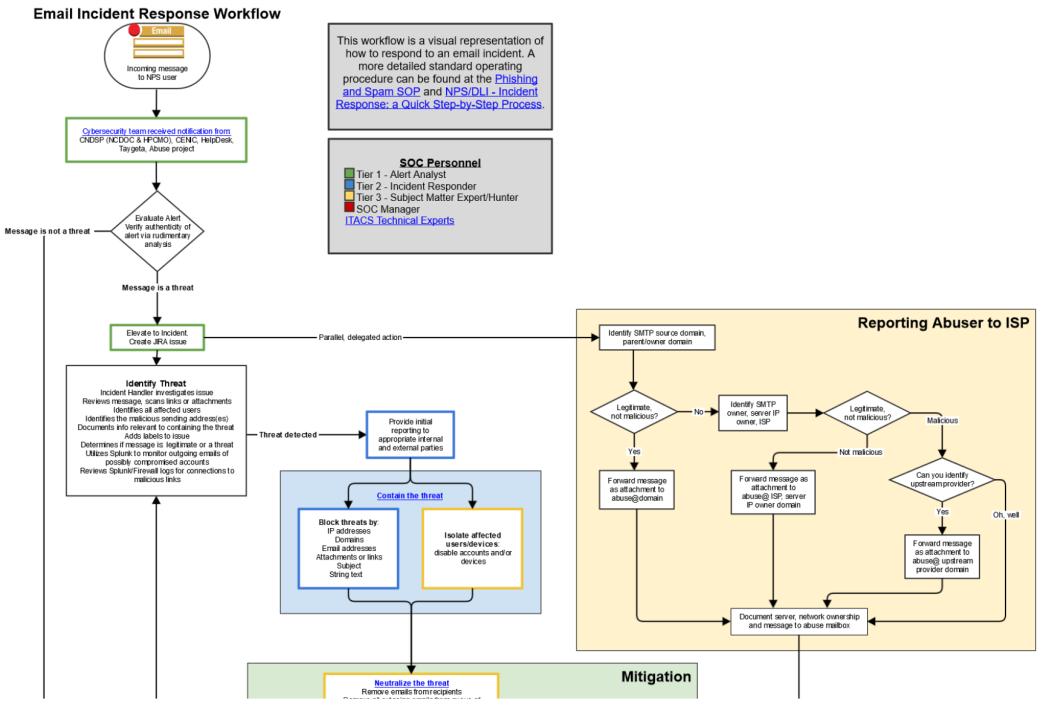


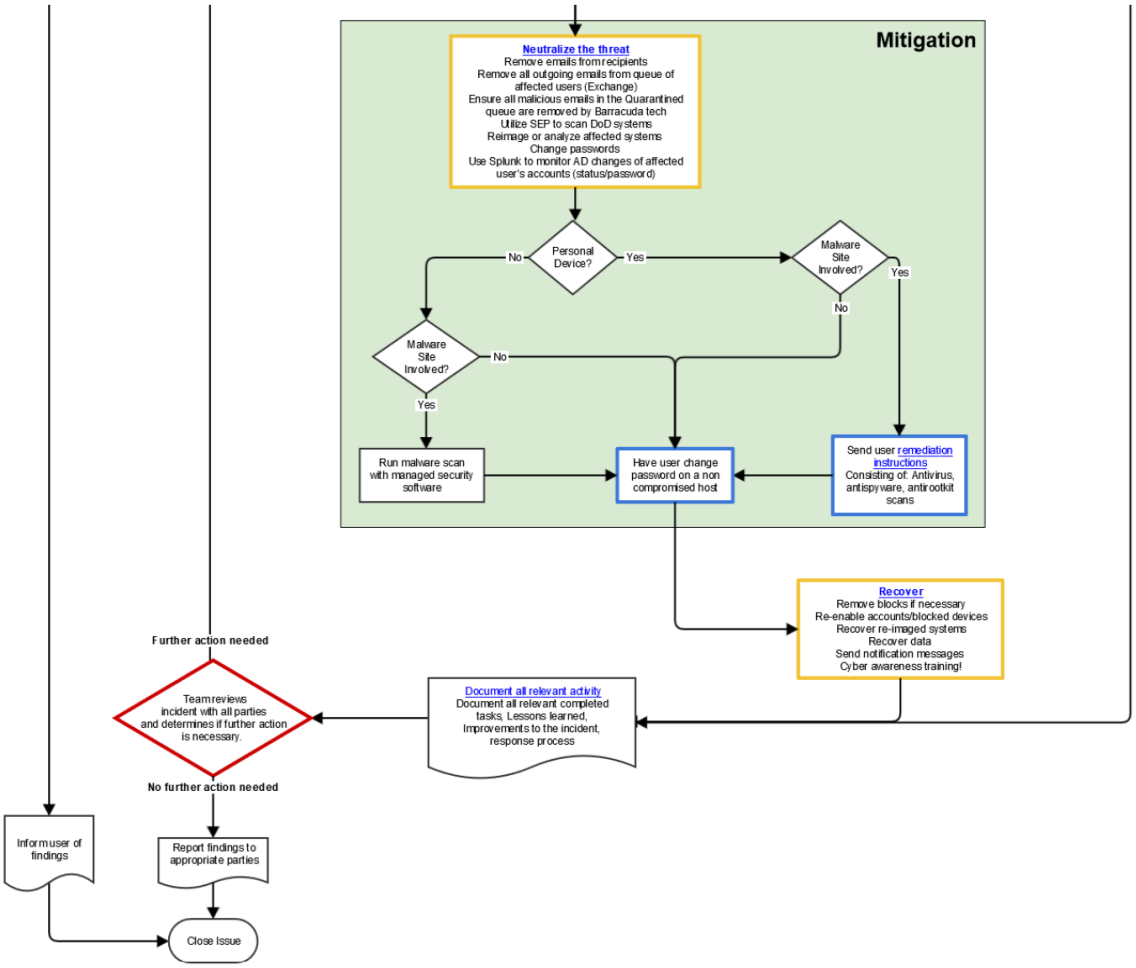

Neutralize the threat
Reimage or analyze affected device(s)
Change passwords
Apply any updates to host
Personal device?
Send user remediation instructions
Consisting of: Antivirus, antispyware, antirootkit scans
Have user change password on a non compromised host
Recover
Remove any blocks if necessary
Re-enable accounts & devices
Recover re-imaged devices
Recover data
Cyber Awareness training
Document all relevant activity
Document all relevant completed tasks
Lessons learned
Improvements to the incident response process
Add any new labels
Setup alerts for future activity via IDS, SIEM
Use IoC's to scan across the network: ACAS, firewall, SIEM
Mitigate any affected systems
Team reviews incident & determines if further action is necessary.
Report findings to appropriate personnel
Further action needed
No further action needed
Close issue

APPENDIX B:
NPS Email Incident Response Work-flow

NPS documentation of email incident response. Used for DevOps process team training.

Email Incident Response Workflow



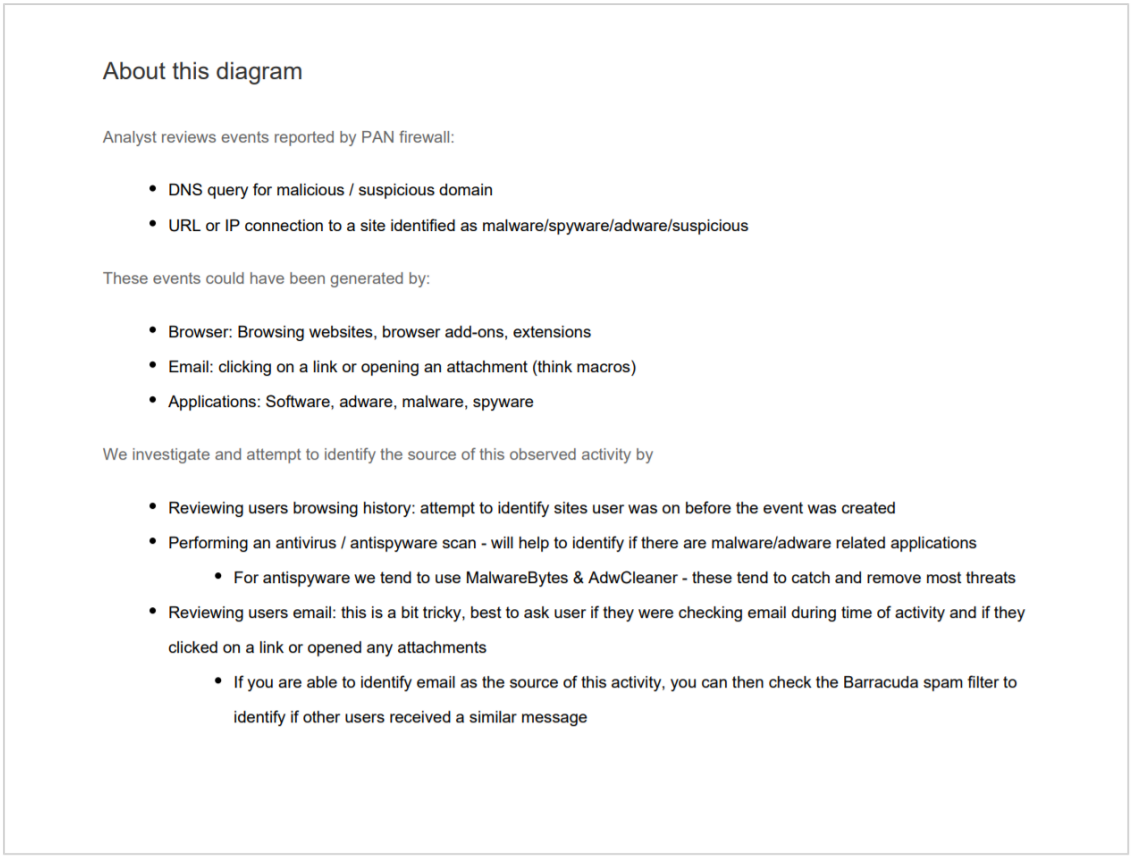

Mitigation
Neutralize the threat
Remove emails from recipients
Remove all outgoing emails from queue of affected users (Exchange)
Ensure all malicious emails in the Quarantined queue are removed by Barracuda tech
Utilize SEP to scan DoD systems
Reimage or analyze affected systems
Change passwords
Use Splunk to monitor AD changes of affected user's accounts (status/password)
Personal Device?
No
Yes
Malware Site Involved?
Yes
No
Malware Site Involved?
No
Yes
Run malware scan with managed security software
Have user change password on a non compromised host
Send user remediation instructions
Consisting of: Antivirus, antispyware, antirootkit scans
Recover
Remove blocks if necessary
Re-enable accounts/blocked devices
Recover re-imaged systems
Recover data
Send notification messages
Cyber awareness training!
Document all relevant activity
Document all relevant completed tasks, Lessons learned, Improvements to the incident, response process
Further action needed
Team reviews incident with all parties and determines if further action is necessary.
No further action needed
Inform user of findings
Report findings to appropriate parties
Close Issue

APPENDIX C:
NPS Malicious URL Response Work-flow

NPS documentation of a malicious URL incident response. Used for DevOps process team training.

Malicious DNS or URL traffic created by browsing, checking email, or applications - identifying the source of the event

What: This diagram was created to be used as a training visual.



About this diagram

Analyst reviews events reported by PAN firewall:

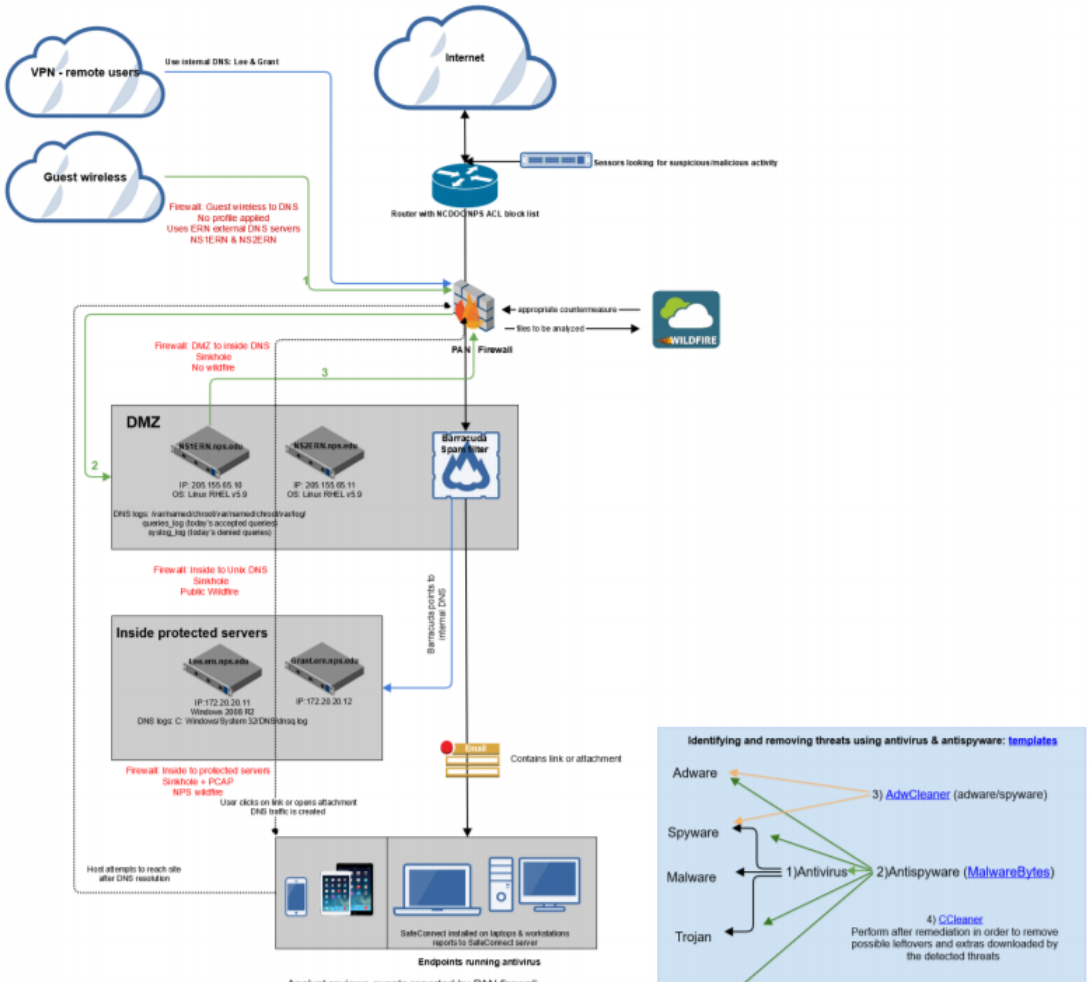
- DNS query for malicious / suspicious domain
- URL or IP connection to a site identified as malware/spyware/adware/suspicious

These events could have been generated by:

- Browser: Browsing websites, browser add-ons, extensions
- Email: clicking on a link or opening an attachment (think macros)
- Applications: Software, adware, malware, spyware

We investigate and attempt to identify the source of this observed activity by

- Reviewing users browsing history: attempt to identify sites user was on before the event was created
- Performing an antivirus / antispyware scan - will help to identify if there are malware/adware related applications
    - For antispyware we tend to use MalwareBytes & AdwCleaner - these tend to catch and remove most threats
- Reviewing users email: this is a bit tricky, best to ask user if they were checking email during time of activity and if they clicked on a link or opened any attachments
    - If you are able to identify email as the source of this activity, you can then check the Barracuda spam filter to identify if other users received a similar message



Analyst reviews events reported by PAN firewall:
DNS query for malicious / suspicious domain
URL or IP connection to a site identified as malware/spyware/adware/suspicious
These events could have been generated by:

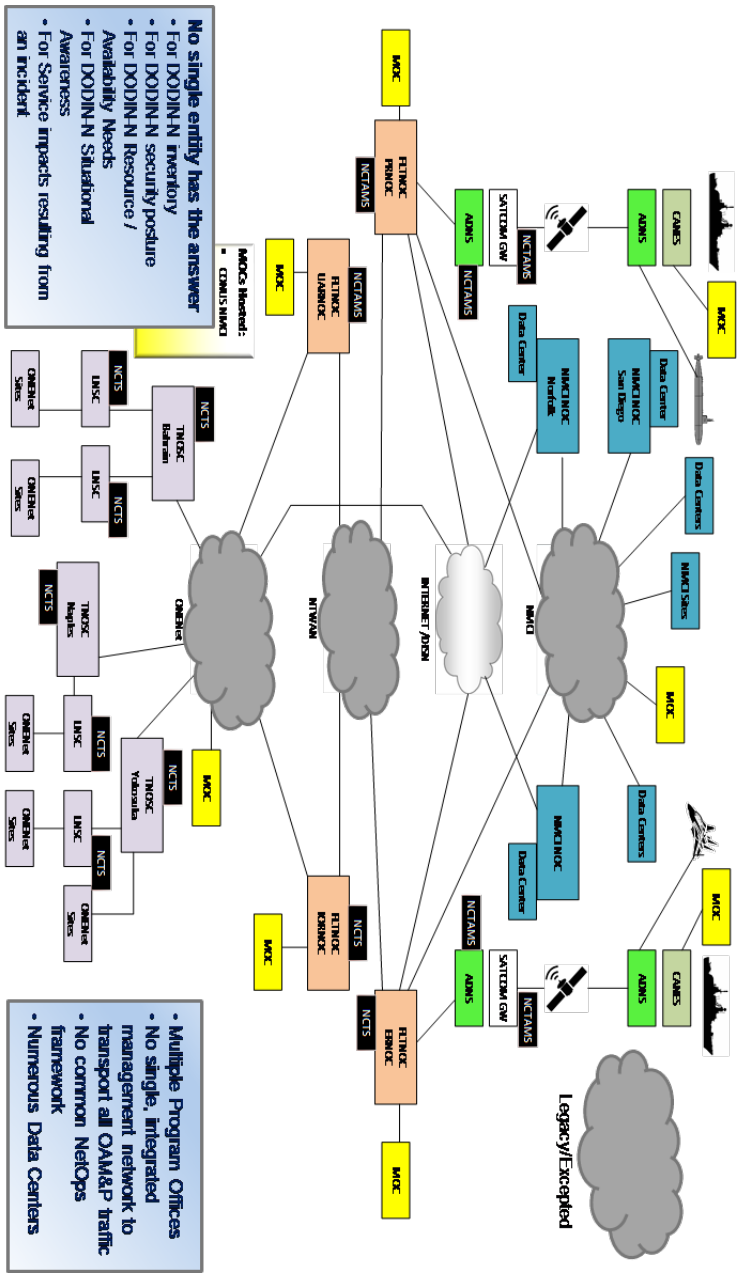
Browser: Browsing websites, browser add-ons, extensions, New browser tabs that load history (thumbnail)

Email: clicking on a link or opening attachment, URLs that load up images

Applications: Software, adware, spyware, malware, Wget, Curl

APPENDIX D:
Current State Of Navy's Network

Current state of Navy network, full size depiction of Figure 3.2 [8]



THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] INOSS IPT, "Integrated Navy Operations Support System (INOSS) Concept of Operations (CONOPS)," U.S. Fleet Cyber Command, March 2019, Draft.

[2] J. M. Richardson. (2018, Dec). Design for Maintaining Maritime Superiority 2.0. [Online]. Available: https://news.usni.org/2018/12/17/design-maintaining-maritime-superiority-2-0

[3] WELCOME TO FCC/C10F. (n.d.). U.S. Fleet Cyber Command. [Online]. Available: https://www.public.navy.mil/fcc-c10f/Pages/home.aspx. Accessed Oct. 31 2019.

[4] Our History. (n.d.). Navy Network Warfare Command. [Online]. Available: https://www.public.navy.mil/fltfor/nnwc/Pages/Command-History.aspx. Accessed May. 16 2019.

[5] Welcome to Navy Cyber Defense Operations Command (NCDOC) Website. (n.d.). Navy Cyber Defense Command. [Online]. Available: https://www.public.navy.mil/fltfor/ncdoc/Pages/default.aspx. Accessed May. 16 2019.

[6] J. Grimes, *NetOps for the Global Information Grid (GIG)*, DODI 8410.02, DOD CIO, Washington, DC, 2012. Available: https://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/841002p.pdf

[7] L. Bass, I. Weber, and L. Zhu, *DevOps: A Software Architect's Perspective*, 1st ed. Westford, Massachusetts: Pearson Education, Inc, 2015.

[8] INOSS IPT, "Integrated Navy Operations Support System (INOSS) Framework," U.S. Fleet Cyber Command, March 2019, Draft.

[9] DoD CIO. (2019, Aug. 12). DoD Enterprise DevSecOps Reference Design. Department of Defense. [Online]. Available: https://dodcio.defense.gov/Portals/0/Documents/DoD%20Enterprise%20DevSecOps%20Reference%20Design%20v1.0_Public%20Release.pdf?ver=2019-09-26-115824-583

[10] J. McQuade, R. Murray, G. Louie, M. Medin, J. Pahlka, and T. Stephens, "Software is Never Done: Refactoring the Acquisition Code for Competitive Advantage," Defense Innovation Board (DIB), Mar. 12 2019, Draft.

[11] NetOps Meets DevOps The State of Network Automation. (2018). *redhat.com*. RedHat. [Online]. Available: https://www.redhat.com/cms/managed-files/ma-state-of-network-automation-analyst-paper-f13966bf-201809-en.pdf

[12] HP Software Services. (2012, Nov.). Analyze performance, improve efficiency HP Network Node Manager iSPI Performance for Metrics software. Hewlett Packard. [Online]. Available: http://www.savli.com/files/docs/hp/nnmi/HP-NNMi-iSPI-Data-Sheet.pdf

[13] *Naval Telecommunications Procedures (NTP) 4 (E)*, Commander, Naval Network Warfare Command, Jan 2008, Restricted Document.

[14] D. Purkiss. (2008, Oct.). Empowering the Information Systems Technician and Information Professional Workforce with an ITIL Framework. [Online]. Available: https://www.doncio.navy.mil/chips/ArticleDetails.aspx?ID=2730

[15] NIWC Pacific Cyber Operations. (n.d.). Naval Information Warfare Center Pacific. [Online]. Available: https://www.public.navy.mil/navwar/NIWC-Pacific/technology/Pages/cyber.aspxx. Accessed Oct. 31 2019.

[16] Working Group Promotes Navy Application Modernization and Accelerated Delivery to Afloat Platforms. (2018, Nov). Space and Naval Warfare Systems Command Public Affairs. [Online]. Available: https://www.navy.mil/submit/display.asp?story_id=107778. Accessed Oct. 31, 2019.

[17] PEO C4I Streamlines Afloat Capability Delivery with DevOps Model. (2017, Oct). Space and Naval Warfare Systems Command Public Affairs. [Online]. Available: https://www.navy.mil/submit/display.asp?story_id=103054. Accessed Oct. 31, 2019.

[18] M. Gilbert, *Artificial Intelligence for Autonomous Networks*, 1st ed. Danvers, MA: CRC Press, 2018.

[19] C. Kershner, Private Communication, Bedminster Township, NJ., Mar. 21 2019, discussion and presentation on infrastructure and automation processes.

[20] M. Robuck. (2019, Aug). ONAP reaches to the sky with satellite deployment by SES Networks. [Online]. Available: https://www.fiercetelecom.com/telecom/onap-reaches-to-sky-satellite-deployment-by-ses-networks. Accessed Dec. 05, 2019.

[21] J. Szewc, "Automation, analytics, and access policies: Edge to the data center to multicloud with extreme management center," private presentation, San Jose, CA., Mar. 23 2018, discussion and presentation on infrastructure and automation processes.

[22] Pricing FAQs. (n.d). Splunk. [Online]. Available: https://www.splunk.com/en_us/software/pricing/faqs.html. Accessed Dec. 05 2019.

[23] J. Thoreson, Private communication, Norfolk, VA, Nov. 24 2019, Email and in person conversation regarding Perspecta organizational roles for Account Services Executives.

[24] T. Wood. (2019, Dec). 5 Stages of ITIL Lifecycle for Services: New ITIL Lifecycle Structure. [Online]. Available: https://blog.masterofproject.com/itil-lifecycle-services/

[25] J. v. d. Hoek. (2020, Jan). What is the Agile Development Cycle? A Quick Intro to Agile Development. [Online]. Available: https://www.mendix.com/blog/pursuing-a-full-agile-software-lifecycle/

[26] B. Gallo. (2016, Dec). Big Data and Analytics: Delivering Insights to the Department of the Navy's Decision Makers. [Online]. Available: https://www.doncio.navy.mil/chips/ArticleDetails.aspx?id=8487

[27] Machine Data Management Analytics: Splunk Enterprise. (n.d.). Splunk. [Online]. Available: https://www.splunk.com/en_us/software/splunk-enterprise.html. Accessed Dec. 05, 2019.

[28] REST API User Manual. (n.d.). Splunk. [Online]. Available: https://docs.splunk.com/Documentation/Splunk/8.0.1/RESTUM/RESTusing. Accessed Dec. 05, 2019.

[29] Frequently Asked Questions. (n.d.). Defense Intelligence Information Enterprise. [Online]. Available: https://www.di2e.net/display/DI2E/DI2E+DevTools. Accessed Dec. 05, 2019.

[30] W. Delores, "Digital Transformation Compile to Combat in 24 Hours," NIWC Pacific Presentation, Apr 2019. Available: https://slideplayer.com/slide/17502480/

[31] B. Gallo. (2018, Sep). Navy Aims for "Compile to Combat in 24 Hours. CHIPS Magazine. [Online]. Available: https://www.doncio.navy.mil/chips/ArticleDetails.aspx?ID=10501

[32] INOSS IPT, "Integrated Navy Operations Support System (INOSS) Gap Analysis and Network Operations (NetOps) Assessment," U.S. Fleet Cyber Command, March 2019, Draft.

[33] *FACE™ Technical Standard, Edition 3.0*, 3rd ed., The Open Group LLC, Park, Burlington, MA, 2017, pp. 1–9.

[34] S. Simi and W. Wayne. (2016, July 15). Implementing FACE-conformant avionics systems. Military Embedded Systems. [Online]. Available: http://mil-embedded.com/articles/implementing-face-conformant-avionics-systems/

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California