2016-06

Server-based and server-less BYOD solutions to support electronic learning

McCarthy, Brian R.; Benson, Joshua C.

Monterey, California: Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

SERVER-BASED AND SERVER-LESS BYOD SOLUTIONS TO SUPPORT ELECTRONIC LEARNING

by

Joshua C. Benson
Brian R. McCarthy

June 2016

Thesis Advisor: Man-Tak Shing
Co-Advisor: Arijit Das

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

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

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2016	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE
SERVER-BASED AND SERVER-LESS BYOD SOLUTIONS TO SUPPORT ELECTRONIC LEARNING

5. FUNDING NUMBERS

6. AUTHOR(S)
Joshua C. Benson and Brian R. McCarthy

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)
USMC College of Distance Education and Training, MCB Quantico, VA 22134-5118

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. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Over the past 10 years, "bring your own device" has become an emerging practice across the commercial landscape and has empowered employees to conduct work-related business from the comfort of their own phone, tablet, or other personal electronic device. Currently in the Department of Defense, and specifically the Department of the Navy, no viable solution exists for the delivery of eLearning content to a service member's personal device that satisfy existing policies. The purpose of this thesis is to explore two potential solutions: a server-based method and a server-less method, both of which would allow Marines and Sailors to access eLearning course material by way of their personal devices. This thesis will test the feasibility and functionality of our server-based and server-less solutions by implementing a basic proof of concept for each. The intent is to provide a baseline from which further research and development can be conducted, and to demonstrate how these solutions present a low-risk environment that preserves government network security while still serving as a professional military education force multiplier. Both solutions, while demonstrated with limited prototypes, have the potential to finally introduce bring your own device into the Department of the Navy's eLearning realm.

14. SUBJECT TERMS
bring your own device, personal electronic device, server-based, server-less, untethered, eLearning

15. NUMBER OF PAGES
97

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

SERVER-BASED AND SERVER-LESS BYOD SOLUTIONS TO SUPPORT ELECTRONIC LEARNING

Joshua C. Benson
Captain, United States Marine Corps
B.S., United States Naval Academy, 2008

Brian R. McCarthy
Captain, United States Marine Corps
B.S., United States Merchant Marine Academy, 2009

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 2016

Approved by: Man-Tak Shing
Thesis Advisor

Arijit Das
Co-Advisor

Peter Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Over the past 10 years, “bring your own device” has become an emerging practice across the commercial landscape and has empowered employees to conduct work-related business from the comfort of their own phone, tablet, or other personal electronic device. Currently in the Department of Defense, and specifically the Department of the Navy, no viable solution exists for the delivery of eLearning content to a service member’s personal device that satisfy existing policies. The purpose of this thesis is to explore two potential solutions: a server-based method and a server-less method, both of which would allow Marines and Sailors to access eLearning course material by way of their personal devices. This thesis will test the feasibility and functionality of our server-based and server-less solutions by implementing a basic proof of concept for each. The intent is to provide a baseline from which further research and development can be conducted, and to demonstrate how these solutions present a low-risk environment that preserves government network security while still serving as a professional military education force multiplier. Both solutions, while demonstrated with limited prototypes, have the potential to finally introduce bring your own device into the Department of the Navy’s eLearning realm.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ARP	Address Resolution Protocol
BYOD	bring your own device
CAC	common access card
CDET	College of Distance Education and Training
CFR	Code of Federal Regulations
CIO	Chief Information Officer
CMD	commercial mobile device
COTS	commercial off-the-shelf
CPU	central processing unit
CUI	controlled unclassified information
DEE	Department of Defense Enterprise Email
DHCP	Domain Host Configuration Protocol
DISA	Defense Information Systems Agency
DLRC	Deployable Learning Resource Center
DMUC	Department of Defense Mobility Unclassified Capability
DNS	Domain Name System
DOD	Department of Defense
DODIN	Department of Defense Information Network
DON	Department of the Navy
DWWG	Department of the Navy Wireless Working Group
EDCOM	Marine Corps Education Command
FOB	forward operating base
FOUO	for official use only
FTP	File Transfer Protocol
GHz	gigahertz
GSA	General Services Administration
GIG	Global Information Grid
HMAC	hash-based message authentication code
ID	identifier
IP	Internet protocol

IT	information technology
JSON	JavaScript Object Notation
KB	kilobyte
LAN	local area network
LRS	learning record store
MARFORRES	Marine Corps Forces Reserves
MAS	mobile application store
MB	megabyte
MDM	mobile device management
MEF	Marine Expeditionary Force
MOS	military occupational specialty
NIPRNET	Nonsecure Internet Protocol Router Network
NIST	National Institute of Standards and Technology
NMCI	Navy Marine Corps Intranet
NKO	Navy Knowledge Online
NSD	National Security Directive
OS	operating system
OWA	Outlook Web Access
PC	personal computer
PED	personal electronic device
PDA	personal digital assistant
PII	personally identifiable information
PIN	personal identification number
PKI	public key infrastructure
PME	professional military education
PMO	program management office
QR	Quick Response
RMF	risk management framework
SAP	special access program
SCI	sensitive compartmented information
S/MIME	secure/multipurpose Internet mail extensions
SMS	Short Message Service

TB	terabyte
TEM	telecommunications expense management
TPM	trusted platform module
URL	uniform resource locator
USCYBERCOM	United States Cyber Command
WAP	Wireless Application Protocol

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

We would like to thank our thesis advisors, Professors Shing and Das, for their tireless dedication in ensuring we provided the best possible product for the issue at hand. Your guidance and mentorship over the past two years has been invaluable, and we will forever be indebted for it.

We would also like to thank all of our classmates in cohort 368–151 for their friendship, camaraderie, and help over the course of the last 24 months. When faced with a challenge or obstacle, we knew we could always lean on each other and bring the best out in each of us.

Lastly, we would like to thank Chrissa and Kristen for their love and stability during our time in Monterey. While in nearly every other job an employee can be replaced, a committed mother and wife cannot. You two pushed us and encouraged us to complete what we never thought we could do, while still raising three beautiful children. This thesis and degree is just as much yours as it is ours.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Today, almost every single service member owns a personal mobile device in the form of a smart phone, tablet, or combination of the two. These devices have become so common and relied upon that they are practically an extension of the human body. As Aaron Smith notes in his article for the Pew Research Center, 85 percent of Americans between the ages of 18 and 29 own a smartphone (2015, p. 3). In 2014, the Department of Defense (DOD) Demographics report stated that 71.7 percent of our enlisted force is under 30 years of age (Department of Defense [DOD], 2014). Combining these statistics with the fact that military members live away from their home of record and require means of communication with family, friends, and coworkers, one would be hard-pressed to find a service member living off the grid in terms of mobile communication.

The DOD as a whole needs to find a way to successfully leverage the growth in ownership, computing power, and technology associated with mobile devices. These devices can be cell phones, tablets, and even personal computers. In 2012, the DOD Chief Information Officer (CIO) rolled out the DOD Mobile Device Strategy with three distinct goals in mind: "advance and evolve the DOD information enterprise infrastructure to support mobile devices, institute mobile device policies and standards, and promote the development and use of DOD mobile and web-enabled applications" (DOD, 2012, p. 1). These goals focused on increasing the DOD's ability to integrate new technologies that will allow it to become more effective as a military and assist in mission accomplishment across the board.

As a follow-up to the CIO's vision, the DOD Commercial Mobile Device (CMD) Implementation Plan was released in February 2013 to promote the development and use of mobile applications within the DOD enterprise, focusing on the non-tactical aspects of the DOD. The Defense Information Systems Agency (DISA) was tasked with being the lead agency responsible for the development and management of the mobile device

initiative. To date, nothing new has been released providing direction or further guidance for the continued exploration or implementation of a mobile device plan.

B. SCOPE

The College of Distance Education and Training (CDET) is an organization within the Marine Corps Education Command (EDCOM) with the mission "to design, develop, deliver, evaluate, manage, and resource distance learning products for programs across the Marine Corps training and education continuum in order to increase operational readiness" (United States Marine Corps, n.d.). We have partnered with CDET to help examine new concepts in incorporating a Bring Your Own Device (BYOD) strategy for their eLearning environment.

For the purposes of this thesis, we will focus on the overarching BYOD movement toward eLearning specifically. Using personal devices for logistics, planning, and other work-related tasks will not be evaluated. The study will identify policies, guidelines, and instructions that pertain to BYOD, and we will make appropriate recommendations that allow for BYOD to be used for eLearning purposes only. We will then identify and research two different options for allowing Marines to use their personally owned devices to take online courses and conduct annual training. Our first method will focus on a ship-based server that communicates with personal devices for the purpose of completing a course of instruction and houses an offline learning management system for the Marines it services. The second method will explore a server-less method for Marines to download, launch, and submit course material from their personal device without the use of a co-located server. We will identify the application requirements and implement a way for the mobile device application to provide CDET with a course completion message that can be transmitted via text message or email and imported into the learning management system hosted by CDET servers. We will provide a proof of concept for both methods to show that a personal device can download, launch, and transmit scores/results to a specific location where a learning management system can then update records accordingly. We will conclude this thesis with some future research recommendations that can help make a comprehensive BYOD strategy a DOD reality.

C. BRING YOUR OWN DEVICE

The practice of using personally owned electronic devices such as tablets, smart phones, and laptops for work-related purposes has been termed "Bring Your Own Device." This concept has grown rapidly in the private sector due to the advancements in technology, computing power, and proliferation of personally owned electronic devices. If applied properly, BYOD could dramatically help service members complete specified tasks and annual training requirements more efficiently. At the small-unit level, government owned computers are limited in quantity and are usually reserved for senior enlisted leaders and officers. Young Marines are expected to complete online training on their own time due to the limited resources available to them while in garrison. Implementing a BYOD strategy aimed at allowing these Marines to use their own devices for work-related purposes between training events or during grey space in their schedule would increase completion statistics, moral and mission effectiveness. By allowing Marines to accomplish work-related tasks more efficiently throughout the workday, their personal time becomes less overwhelmed with administrative necessities.

ELearning encompasses the closest possible scenario for BYOD implementation in the military because of the current online format, accessibility and limited operational impact. By starting out with a BYOD plan focused on academics and training, the government can further analyze the impact a full-scale BYOD strategy will have on other more sensitive operational activities.

D. LIMITATIONS AND CONCERNS REGARDING BYOD

One of the biggest concerns with BYOD is security. By allowing service members to use their personal devices to access privileged or secured networks, the possibility of data breaches, leaks and information assurance issues does arise. To limit this risk, it is imperative to have a well-developed and organized security policy in place that supports and monitors BYOD use. According to Rob Anderson, the Marine Corps' vision and strategy division chief, the Marine Corps is seeking to outsource mobile security requirements to commercial carriers such as Sprint, Verizon and AT&T; a prime

example that the DOD is making a conscious effort to try to integrate emerging electronic technologies into the government workplace (Grim, 2013).

The DOD as a whole requires a specific method of authentication to be used for all government websites. The common access card (CAC) in combination with a personal identification number (PIN) is used to verify identity and allow access to government networks. Some government entities have been given waivers to this policy to allow access with a correct username and password. For BYOD to be successful in the military, the username and password login must become standard for all BYOD-accessible services. Mobile CAC readers do exist, but the cost associated with supplying these for every service member is prohibitive. It is imperative that username and password login be used in the BYOD eLearning environment.

II. LITERATURE REVIEW

The purpose of this research was to identify potential methods for the Department of the Navy (DON) to implement a BYOD strategy aimed at the eLearning environment for Marines and Sailors, both forward deployed and in garrison. Before developing our proofs of concept to demonstrate a BYOD-friendly eLearning solution, it is important to review the current DOD and DON instructions, policies, and guidelines concerning mobile devices, official websites, and security requirements in order to fully understand the strategic visions for the future.

A. DOD INFORMATION SECURITY POLICIES

In February 2012, the DOD released their four-volume *Information Security Program* manual, (*DOD Manual 5200.01*), which would look to

> implement policy, assign responsibilities, and provide procedures for the designation, marking, protection, and dissemination of controlled unclassified information (CUI) and classified information, including information categorized as collateral, sensitive compartmented information (SCI), and Special Access Program (SAP). (DOD, 2012, p. 1)

While the scope of this thesis does not delve into the classified realm, the groundwork for a deployable BYOD solution must be centered on this document, specifically Volume 1 and Volume 4. Volume 1 outlines and defines the DOD Information Security Program as a whole, providing the necessary guidance and direction for the enterprise moving forward, while Volume 4 gives instruction for the identification and protection of CUI. The same standards that apply to a Marine or Sailor accessing professional military education (PME) materials online via their personal computer or laptop will undoubtedly have to also apply in a BYOD environment. All classes and materials made available in the proposed BYOD environments would still have to meet the standards outlined in DOD Directive 5230.9 for Clearance of DOD Information for Public Release. The preponderance of eLearning typically occurs on an approved Non-secure Internet Protocol Router Network (NIPRNet) computer, or a service member's personal computer or laptop, so accommodating a BYOD environment through this

policy should not prove too daunting. The service member would still have the ability to access the material via a non-DOD wireless or cellular network, or by pulling the material from an untethered, dedicated eLearning server. The security of the network would still be maintained, the material could still be controlled by way of the respective service's learning network (MarineNet, Navy Knowledge Online (NKO), etc., all while making the material readily available for use on their personal mobile device.

B. COMMERCIAL MOBILE DEVICE IMPLEMENTATION PLAN

In this section we will explore the detailed nuances of the CIO's CMD Implementation Plan released over three years ago, and its role in the development, implementation, and employment of BYOD in the DOD.

1. Overview

On 15 February 2013, the then CIO for the DOD, Teresa Takei, released the DOD CMD Implementation Plan that was aimed at providing a phased agenda that would lead to the introduction, integration, and development of mobile non-tactical applications throughout the entire DOD (DOD, 2013). With the ever-increasing reliance on commercial off-the-shelf (COTS) mobile devices, to include smart phones, tablets, and personal laptops, it has become a priority of the DOD to try to leverage these assets to better enable and facilitate warfighters and planners across the full spectrum of operations. The plan's specific focus, and overall end state, will be to, in phases, integrate mobile technology in day-to-day military operations, both secure and non-secure, and eventually implement a cloud-enabled command and control capability that would serve as a force multiplier and allow full-scale collaboration of information (DOD, 2013). Although not an integral aspect of the overall plan, BYOD and its increasing popularity in the private sector, is addressed along with limited and open-ended guidance provided for future considerations.

2. Vision

Ultimately, the CIO's vision for the DOD CMD Implementation Plan is to allow for the integration of cutting-edge CMDs, to include their infrastructure, and to allow for

personnel of the DOD to be interconnected globally. However, the DOD believes that the path to achieving this goal will require an incremental, cost-effective plan that explores all feasible commercial solutions and does not focus on one particular type of mobile technology. DOD leadership emphasizes that building this infrastructure will require planners and subject matter experts to leverage the operability and capability of CMDs, while not compromising DOD interests or information, especially in the classified domain.

Due to the fact that there currently is no multi-level security in place for mobile device management (MDM) systems, the DOD mandates that separate systems will be required for both the classified and unclassified domains. Generally speaking, there is greater flexibility in developing and providing a MDM capability to DOD unclassified networks, but a "centralized methodology" will be needed for all classified networks (DOD, 2013, p. 4). Additionally, the CIO's intent is to develop an overarching governing process to regulate mobile application integration due to the expeditious nature in which mobile applications are developed in present day. The end goal will be to construct and maintain a secure, centralized library to allow users to access vetted mobile applications for their device for free at any time (DOD, 2013).

3. Approach

The approach to building a robust, effective CMD DOD enterprise will be a "continuous process" that will look to evaluate requirements and various business case analyses in an effort to best identify the overarching mission, while also determining cost effectiveness (DOD, 2013, p. 4). In a collective effort, the CIO states that DISA, various DOD components, and the General Services Administration (GSA) will head the procurement and operation of CMDs being considered for the plan. To accomplish this, the DOD believes that a myriad of different approaches will be required in order to expeditiously satisfy the broad scope of DOD applications, while still harboring a competitive acquisition environment, implementing mobile services interoperability, and furthering the adoption of emerging technologies.

The CMD Implementation Plan's approach encompasses four key guiding features that the CIO deems critical to the future success of the enterprise. The first is governance, which will look to establish a process by which standards, policies, and processes are developed to manage mobile applications for the DOD. The second tenant outlined by the CIO, centralized enterprise implementation, states that "DISA shall establish a DOD Mobility Program Management Office (PMO) that will provide guidelines for secure classified and unclassified mobile communications capabilities to the DOD on a global basis" (DOD, 2013, p. 6).

The third feature of the CIO's approach, DOD Components implementations, is an effort to establish and manage a MDM and/or Mobile Application Store (MAS) system, which would meet pre-validated DOD mobile device requirements, to support DOD mobile users. Per the CMD Implementation Plan, an MDM or MAS can be created for initial operational uses, but ultimately must be integrated into the overarching DOD enterprise by way of the prescribed convergence plan. Additionally, the plan states that all systems must meet DOD security requirements, and network management information for each system must be reported through the proper channels using the current United States Cyber Command (USCYBERCOM) guidelines. Table 1, referenced directly from the CIO's plan, outlines a series of component mobility pilots and initial operational uses that will be assessed to determine their impacts on current DOD mobility services.

Finally, the last key tenant explained by the CIO, GSA implementation, looks to empower the administration to establish contract vehicles for devices and wireless services, construct a MDM platform for device monitoring, security, and management, as well as update dot gov domain catalogues, guidance, and procedures in order to support agencies within the enterprise. Per the DOD plan, mobility services contracts for DOD Components from the GSA can be explored only once the MDM/MSA has met the pre-defined DOD-level security requirements. Evaluation of GSA-developed MDM/MSA systems is continuous, and the DOD will assess each individual solution as it is made accessible (DOD, 2013).

Table 1. DOD Component Mobility Pilots. Source: DOD (2013, p. 8).

Unclassified CMD Capability	Classified CMD Capability
• Army App Store (USA) • Connecting Soldiers to Digital Apps (CSDA) (USA) • Digital Sea Bag (USN) • Warfighter's Edge (Wedge) (USAF) • Electronic Flight Bags (USAF) • ONE Mobile Application (USNORTHCOM) • mCARE Initiative (USA/TATRC) • 92Y Instructor (USA/TRADOC) • Fixed Wireless at a Distance (DARPA)	• 4G/LTE Sea Trial (USN) • SECRET BlackBerry (USSOCOM) • Trusted Handheld (USMC) • Secure iPad (SiPAD) (DARPA) • Multi-Level Security (MLS) Joint Capability Technology Demonstration (JCTD) (DISA) • JO-LTE-D TACTICS JCTD (DISA) • TIPSPIRAL (NSA)

4. Implementation Framework

The CIO's implementation framework provides initial, basic guidance to manage the infrastructure, devices, and applications associated with the integration of CMDs in the DOD. Her initial intent, when the DOD published the CMD Implementation Plan in February 2013, was to create additional DOD policy, provide more information, and assimilate feedback and lessons learned from initial enterprise mobile implementation by March 2013.

In addition to infrastructure, devices, and applications, the CIO states that the framework would also yield direction on cost management, with the primary goal to provide the best mobile solutions to meet mission requirements while obtaining the best value for the DOD. Per the DOD, the acquisition contracts for carrier services, both voice and data, for approved CMDs will be consolidated to minimize the waste of government resources. Additionally, the CIO further notes that all carrier accounts will be administered and overseen using a telecommunications expense management (TEM) system that will manage any underutilized or over-subscribed accounts to avoid unnecessary spending.

All devices will be supported by a multi-vendor mobile operating system (OS) environment to allow for a streamlined, "device-agnostic procurement approach" to deter commercial favoritism of one specific device's operation or performance over another

(DOD, 2013, p. 9). The framework also explains that all mobile applications will first be approved by way of a strict governance process to ensure network security and enterprise applicability, and will then be stored and distributed in an application development framework that will allow for interoperability across many OSs.

The CIO emphasizes that the intent of the framework is to incorporate cutting-edge commercial capabilities and further the use of standards while maintaining security requirements compliance. Her overarching intent is to have the CMD security approval process take no more than 90-days for all mobile devices and operating systems. Additionally, application development initiatives and pilot demonstrations, per the CIO's guidance previously published in the CMD Interim Policy as well as her memorandum on the use of CMDs in the DOD, must conform to the security guidelines and receive approval before accessing any DOD network.

Per the DOD, MDM systems will conduct routine and in-depth remote scanning in order to ensure compliance across the enterprise. The CIO also states that unclassified and classified material being processed on their respective medium using CMDs will operate in accordance with their corresponding DOD instructions and directives, as well as National Security Directive (NSD)-42. Each individual DOD component will train all DOD members on the regulations and responsibilities associated with the use of CMDs in the workplace (DOD, 2013).

5. Procedures

Following are the CIO's intentions toward exploring BYOD as a solution moving forward for DOD mobility. Only the BYOD portion is included due to the scope of this thesis.

a. Bring Your Own Device

The CIO identifies how implementing a BYOD solution could offer many "compelling benefits to organizations and users" in the long term (DOD, 2013, p. 16). However, she addresses the fact that "current DOD policies, operational constructs, and security vulnerabilities" inhibit the ability to allow service members to associate their

own personal devices with government networks (DOD, 2013, p. 16). Essentially, if the device is not procured through the proper DOD acquisition channels and vetted by lawmakers, then it does not belong on a government network. When the CMD Implementation Plan was drafted in 2013, an approach was identified to develop "hardened" devices that utilized a Trusted Platform Module (TPM) that was capable of storing cryptographic keys, or other implementations referred to as TPM chip or TPM Security Device, that could allow a device to toggle between enterprise and personal use (DOD, 2013, p. 16). However, this technology will only realistically be explored once stringent DOD security requirements are met, and the DOD has furnished the requisite policies to support BYOD use on a government network. The CIO concludes by stating that the DOD "will continue to evaluate BYOD options" in concurrence with the Digital Government Strategy (DOD, 2013, p. 16). Simply put, BYOD does not exist in the DOD, and it does not appear that it will for the foreseeable future.

C. DEFENSE INFORMATION SYSTEMS AGENCY'S ROLE

The Defense Information Systems Agency (DISA) serves as the conduit between the commercial information technology (IT) landscape and the DOD. Their overall mission objectives include command and control synchronization across the DOD, reducing duplication costs that may occur in production and operations, and enabling the warfighter through a robust telecommunications infrastructure that facilitates global information sharing across the enterprise (Defense Information Systems Agency [DISA], 2015). In 2012, the CIO released the DOD Mobile Device Strategy that intended to "align the progress of various mobile device pilots and initiatives across DOD under common objectives, ensuring that the warfighter benefits from such activities" (DOD, 2012, para. 1). This document sought to begin the process of integrating mobile devices into the DOD communication enterprise, while still adhering to the strict policies in place.

Per their official government website, one of DISA's largest continuing efforts to date has been the DOD Mobility Unclassified Capability (DMUC); a service that grants government purchased, formally acquired CMD access to the Department of Defense Information Network (DODIN), Defense Enterprise Email (DEE), encrypted email

capability, chat, as well as access to several hundreds of approved Apple and Android mobile applications. According to DISA, the overall intent of the DMUC is to no longer have respective Combatant Commands and Agencies bear the burden of developing and managing an infrastructure that allows them connect their CMD to the network. Additionally, by DISA taking charge of this endeavor, a more robust and capable network can be employed to accommodate multiple approved mobile devices and promote global information sharing. As technology and carrier services evolve and become more capable, DISA mentions that they continue to interact with mobile provides, such as Verizon; AT&T; Sprint; and T-Mobile, and mobile operating system vendors, such as Apple; Android; and Windows, to ensure that government procured mobile devices have the latest information to best suit the warfighter. DISA explains that the benefits to the DMUC's MDM endeavor is that it can provide the ability to enforce policy through various user permissions, support virus countermeasures like malware detection and remote data-wipe capabilities, and monitor; secure; and manage the mobile devices on a wide swathe of differing DOD communication environments. It is very apparent that the DOD will only allow enterprise-issued mobile devices that have been vetted through the government acquisitions and program management process to gain access to their network (DISA, n.d.).

D. NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY GUIDE TO GENERAL SERVER SECURITY SPECIAL PUBLICATION 800–123

In 2008, the National Institute of Standards and Technology (NIST) published their guide to general server security that identifies the framework needed for successful employment, maintenance, and configuration of network servers, with Federal agencies as their primary audience. NIST believes that the guide aims to assist responsible system and security administrators in effectively maintaining the security of servers that can provide a wide swathe of services to service members and government employees both deployed and in garrison. Because almost all network assets in the federal government operate on Microsoft Windows, the publication primarily addresses servers that use said operating system, but NIST also offers insight on devices that may use Unix and Linux as well. From a high-level perspective, NIST is able to provide detailed, technical common

practices that should be applied to servers in order to ensure the operating system, software, and appropriate application patches and upgrades are installed, configured, and secured properly.

NIST identifies and recommends five guidelines that every organization should execute when setting up their respective server. The first guideline states that "organizations should carefully plan and address the security aspects of the deployment of the server" (Scarfone, Jansen, & Tracy, 2008, pp. ES-1). The second guideline is that organizations should only apply the appropriate level of security management practices and controls for operation and maintenance of the server in its given environment. The third and fourth guidelines recommend that organizations should make sure the server's operating system and application, respectively, are properly deployed, configured, and maintained to meet the security requirements deemed important by said organization. Finally, the fifth guideline that NIST mentions is that organizations should be vigilant and committed to the ongoing process of server maintenance and security.

While the majority of this publication mainly applies to servers that are, or will be, connected to the broader, in our case NMCI, communications network, all of the aforementioned guidelines would still apply to the untethered, server-based BYOD solution that a portion of this thesis brings to light. Even with an isolated BYOD web server, properly educated system administrators must carry out and execute the necessary, pre-defined network security measures set forth by their given unit or organization. The security policies and procedures related to the server-based BYOD solution are not within the scope of this thesis, but will undoubtedly need to be identified and addressed in a detailed, technical manner for successful employment (Scarfone, Jansen, & Tracy, 2008).

E. DOD CYBERSECURITY INSTRUCTION 8500.01

In March 2014, the CIO released DOD Directive 8500.01, Cybersecurity Instruction, that aimed to identify and establish a robust cybersecurity program to protect and defend DOD information and IT (DOD, 2014). The instruction addresses both classified and unclassified DOD information in electronic format, to include SAP

information and the handling of SCI material. Additionally, the directive formally adopts the term “cybersecurity,” replacing “information assurance” (IA), as the term to be used by the DOD moving forward (DOD, 2014, p. 1).

The CIO determined eleven pieces of critical DOD policy that will procedurally and systemically be implemented across the enterprise in order to ensure the safest and most secure cyber environment possible. The CIO’s eleven policies are risk management, operational resilience, integration and interoperability, cyberspace defense, performance, DOD information, identity assurance, IT, cybersecurity workforce, and mission partners. While all of these policies are extremely important, a select few have been identified as being especially critical to the successful employment of a BYOD solution.

The first, risk management, is important because of the uncertainty surrounding a DOD BYOD-friendly environment. Therefore, prior to introducing BYOD solutions to the DOD, the CIO believes that an in-depth cybersecurity risk management process must be developed in order to protect U.S. interests and DOD assets across the network landscape. Even though one of the proposed solutions in this thesis implements an untethered, isolated local web server, documentation and procedures must still be drafted during the acquisition phase in order to mitigate a plethora of cybersecurity issues.

The CIO’s second prescribed policy, operational resilience, is especially important to a BYOD approach because network administrators will always need the requisite information to troubleshoot or quarantine a particular DOD asset should a compromise occur. Per the directive, a DOD device must “have the ability to reconfigure, optimize, self-defend, and recover with little or no human intervention” (DOD, 2014, p. 3). The CIO contends that this ensures a detailed audit trail is produced that could help a system administrator or contractor diagnose a security threat and prevent further harm.

Lastly, the third policy that is important for a BYOD solution is IT. Per the DOD directive, any piece of IT equipment that receives, processes, stores, displays, or transmits any type of DOD information, to include entry-level PME classes, must be handled and maintained in accordance with cybersecurity policies and standards. Because the solutions in this thesis are geared and tailored toward eLearning environments that

strictly deal with for official use only (FOUO) material, the handling of classified information will not be addressed. Identity assurance, from a BYOD eLearning perspective, will not be discussed because there is no required CAC login procedure when accessing unclassified PME material. To access the materials, whether on the local web server or directly on the user's device from the cellular network, only a pre-established username and password will be required (DOD, 2014).

F. RISK MANAGEMENT FRAMEWORK FOR DOD IT DOD INSTRUCTION 8510.01

DOD Instruction 8510.01, titled *Risk Management Framework for DOD IT*, was released in March 2014 by the CIO in order to outline policy for the DOD to "establish and use an integrated enterprise-wide decision structure for cybersecurity risk management (the RMF)" (DOD, 2014, p. 2). Per the instruction, cybersecurity requirements, previously touched upon earlier in this chapter, for DOD assets and information technologies will be managed through the risk management framework (RMF) in tandem with the principals identified and established by NIST. The issue that continues to surface with the majority of the referenced DOD Instructions published by the CIO, is that they all fail to specifically address policies for mobile devices attached to the enterprise. Because BYOD has not been introduced to the DOD, these policies, some dated over two years ago, have hardly any applicability to a proposed BYOD environment. Even government procured mobile devices appear to be glossed over in this proposed RMF, as well as previous instructions. For a successful BYOD-friendly eLearning network to be fostered, there will need to be detailed policies in place dedicated solely to such a network. Service members who associate their personal mobile devices to an untethered DOD local area network will have to be treated differently than a government approved, issued mobile device. From a policy perspective, a BYOD network and architecture must be treated and addressed completely different than that of a dedicated DOD network (DOD, 2014).

G. DON CIO MEMO, 27 FEBRUARY 2012

The DON CIO's 2012 memo, titled *DON Plan for Optimizing Use of Employee IT Devices and Other IT to Achieve Efficiencies*, explains how various tools were used by commands globally to "track zero-use devices, minute optimization, air card costs, and roaming costs down to the individual level" (Department of the Navy [DON], 2012). The CIO further clarifies that the metrics to be used to enforce compliance of these measures across the DON were, at the time, being finalized, and that new methods and processes would be explored to eliminate any fraud, waste, and abuse on government procured mobile devices. The CIO's message is one that emphatically articulates the DON's cost-saving measures, but it only applies to government-issued, properly acquired mobile devices.

While the expectation to farm out computational power for government-related business to users' personal devices is completely unreasonable, there does however need to be a cost and analysis model created to show how a BYOD learning environment could save the government a substantial amount of money. Rather than spend several hundred dollars per small-unit to field four to five NMCI computers, a unit could be issued one wireless web server that could provide learning content to as many Marines and Sailors that could feasibly connect to it without degrading performance. It would become a force multiplier, and also cut down on operational costs. Additionally, a server-less solution would allow for a Marine or Sailor to simply download the material when in cellular or network range, and then complete the course without any dedicated NMCI asset necessary. A BYOD learning environment would most certainly save money against the fiscal year budget in the long-term (DON, 2012).

H. DON ENTERPRISE MOBILITY, 2008

In 2008, Robert J. Carey, the then DON CIO, released the DON's *Enterprise Mobility* publication which closely mirrors the DOD's CMD Implementation Plan that would be released five years later. His message for the enterprise mobility is that "the net-centric environment of the Global Information Grid (GIG) will provide our warfighters with Information Superiority, affording decision superiority, tactical

advantages and enabling mission accomplishment" (DON, 2008, p. 1). The aim is to provide a fully integrated, synchronous network capability that can provide information to any Sailor or Marine in any clime or place at any given time. Mr. Carey explains that a significant portion of this vision moving forward is to integrate commercial wireless products, such as personal digital assistants (PDAs), BlackBerrys, smart phones, and RFID systems, into the overall enterprise to help provide standardization, real-time access, and system interoperability.

The publication goes on to explain that several working groups were created to develop a collaborative framework moving forward to identify the risks, cost benefits, and feasibility of complete CMD integration across the enterprise. Specifically, a DON Wireless Working Group (DWWG) was formed to help identify gaps, develop and change policies, coordinate efforts, and re-classify requirements moving forward for wireless integration across the DON. Much like the DOD CMD Implementation Plan, there is no plan or vision moving forward for the introduction of BYOD into the DON.

What seems to be continually overlooked in these publications regarding a "net-centric" enterprise is material that, while not strategically important to a given mission or theater requirement, would prove to be ideal for access in a BYOD environment. It is foolish to think that a service member's un-vetted, unsecure mobile device could be fully integrated into the DON mobility enterprise. However, small, untethered, pocket networks could be employed to allow multiple Marines and Sailors to access the necessary military education material they need to succeed both personally and professionally. Additionally, material could be made available for download to their personal device and taken with them to be viewed at any time completely devoid of network connectivity. These are ideas that need to be incorporated in guidance moving forward (DON, 2008).

I. DON SECURITY GUIDANCE FOR PERSONAL ELECTRONIC DEVICES (DON CIO MESSAGE DTG: 202041Z AUG 07)

On August 20, 2007, the DON CIO released his guidance pertaining to the security for personal electronic devices (PEDs) across the enterprise. In the guidance, it

references DOD Instruction 8520.2, titled *Public Key Infrastructure and Public Key Enabling*, that "requires that all DOD information systems, including networks and email systems, be enabled to use DOD issued public key infrastructure (PKI) certificates to support authentication, access control, confidentiality, data integrity, and nonrepudiation" (DON, 2007, p. 1). The guidance explains that as technology has evolved, the use of encryption and digital signatures has become much more common practice, especially with the preponderance of information on DOD networks that contains personally identifiable information (PII) and other sensitive information. Because of this, the DON CIO has made it clear that all PEDs "must be capable of supporting digital signature and encryption (secure/multipurpose Internet mail extensions (S/MIME)) functionality" (DON, 2007, p. 1). It continues by stating that all PEDs "must be able to interface with the PKI certificates stored on DOD-approved hardware tokens including CAC" (DON, 2007, p. 1).

The various types of access control methods that are employed across the DOD will be explored in more detail later in this thesis, but this guidance does not appear to be tailored to an eLearning environment or an audience that is simply trying to access benign learning materials for professional and military education. Due to the fact that the majority of the courses and materials on Marine Net, or any other military learning suite, are made available to a service member's dependents, there is no steadfast requirement for a CAC-enabled authentication process or the use of PKI certificates. Therefore, while this guidance appears to be directed toward Marines and Sailors who plan to use their PEDs for Outlook Web Access (OWA) and other mission-oriented, job-specific duties, it is not all-encompassing and does not directly correlate to a BYOD-friendly eLearning environment and network architecture. Only a valid, issued username and a strong password are required to gain access to the learning environment's materials.

III. SERVER-BASED BYOD SOLUTION

A. SYSTEM DESCRIPTION

Whether deployed or in garrison, Marines and Sailors are constantly navigating Marine Net or NKO courses in order to meet pre-deployment training requirements, better posture themselves for promotion, or pursue an academic topic that may interest them. Rather than cycle fifteen individuals through two dedicated Navy Marine Corps Intranet (NMCI) computers, what if each service member could wirelessly access the very same materials from their personally owned mobile devices?

In present day, individuals can purchase wireless personal servers for a couple hundred dollars that can be set up locally at their house, allowing for friends and family, given access, to view and retrieve media and pictures that are stored on the hard drive. Why can we not replicate this architecture for Marines and Sailors to access PME materials during periods of down time? If someone can set up and configure their own standalone server on their own private network using purely open source materials, why can we not employ this in the bowels of a ship or on a forward operating base (FOB) in Afghanistan? This chapter will lay the framework for future research and development on a possible server-based BYOD solution that can be tailored to a specific unit's needs and requirements, and be implemented in any clime or place to meet personal and unit training objectives.

The end state of the server-based solution is to provide an untethered, tailored, and dedicated wireless asset to Marines and Sailors in any environment to allow for completion of a plethora of different modules available to them on the Marine Corps' Marine Net and the Navy's NKO. The term "untethered" simply means that the proposed server would be isolated and not connected, in any capacity, to any type of DOD or DON network. By being untethered, this would save precious bandwidth, especially when confined to a ship's network or when forward deployed, while also segregating all personal devices to one local network to prevent possible malicious software from gaining access to the overall network. If at any time the server becomes compromised

due to malicious code, it can be quarantined, wiped, and ultimately recycled again without any negative impact on the overall enterprise. The mobile server would be strictly wireless allowing for a wide array of users to gain access to the materials stored on the local server just so long as their personal device is within range and meets pre-defined security authentication procedures.

In order to accommodate Marines and Sailors in any operational environment, the server will be ruggedly designed to withstand normal wear and tear of the climatological surroundings any unit may endure, much like any resident network server would. Any maintenance or troubleshooting on the server can be conducted by a qualified system administrator, presumably an individual who holds a Marine Corps communications military occupational specialty (MOS) or Navy communications specialty code. Prior to a deployment, or even while in garrison, a dedicated system administrator could attend a course on how to properly maintain the server, and also who to contact should an unsolvable issue arise.

When a Marine or Sailor completes a course using their personal device, the server will log that information locally in a learning record store (LRS), and the database will be updated accordingly once the server is connected to a dedicated computer for offload. Ultimately, the goal of the deployable, wireless mobile server is to act as a force multiplier and encourage Marines and Sailors to access materials that are readily available to them using their own devices. Hopefully this solution can foster a better learning environment, save a considerable amount of money and bandwidth, and bring a true BYOD environment to the DOD that has long been a staple in the commercial world.

B. PREVIOUS TRIALS

The Marine Corps has attempted to implement a deployable eLearning solution in the past. According to CDET, the Deployable Learning Resource Center (DLRC) was fielded to all three Marine Expeditionary Forces (MEFs) and the Marine Forces Reserve (MARFORRES) in the hopes of providing a dedicated, satellite learning environment that could meet the many evolving PME requirements faced by Marines both deployed and in garrison. The DLRC, in its simplest form, was a local area network (LAN) consisting of a

single server, which hosted the appropriate PME and eLearning material, and 20 laptop computers that a number of different Marines could log on to in order to access the provided material. Figure 1 illustrates how a conventional DLRC was employed, and also outlines where the 47 total units were fielded across the Marine Corps. However, the DLRC had several issues and limitations that resulted in its cancellation as an eLearning program of record. According to Mr. Dennis Chinault, CDET Operations Officer, the DLRC was much too heavy to be fielded (125 pounds), was entirely too complex to set up when fielded (Mr. Chinault notes that it required an engineering degree to do so), had too large of a footprint (four to five transit cases), and was never able to be deployed on ship due to space limitations. The DLRC was intended to replicate an Internet café atmosphere that fostered a healthy learning environment and facilitated the completion of critical training requirements by dozens of Marines. Unfortunately, its implementation was deeply flawed and the value added was not enough to justify its employment. However, the concept of establishing a LAN, a wireless one in our case, that hosts eLearning material is something that can be built upon when attempting to introduce BYOD.

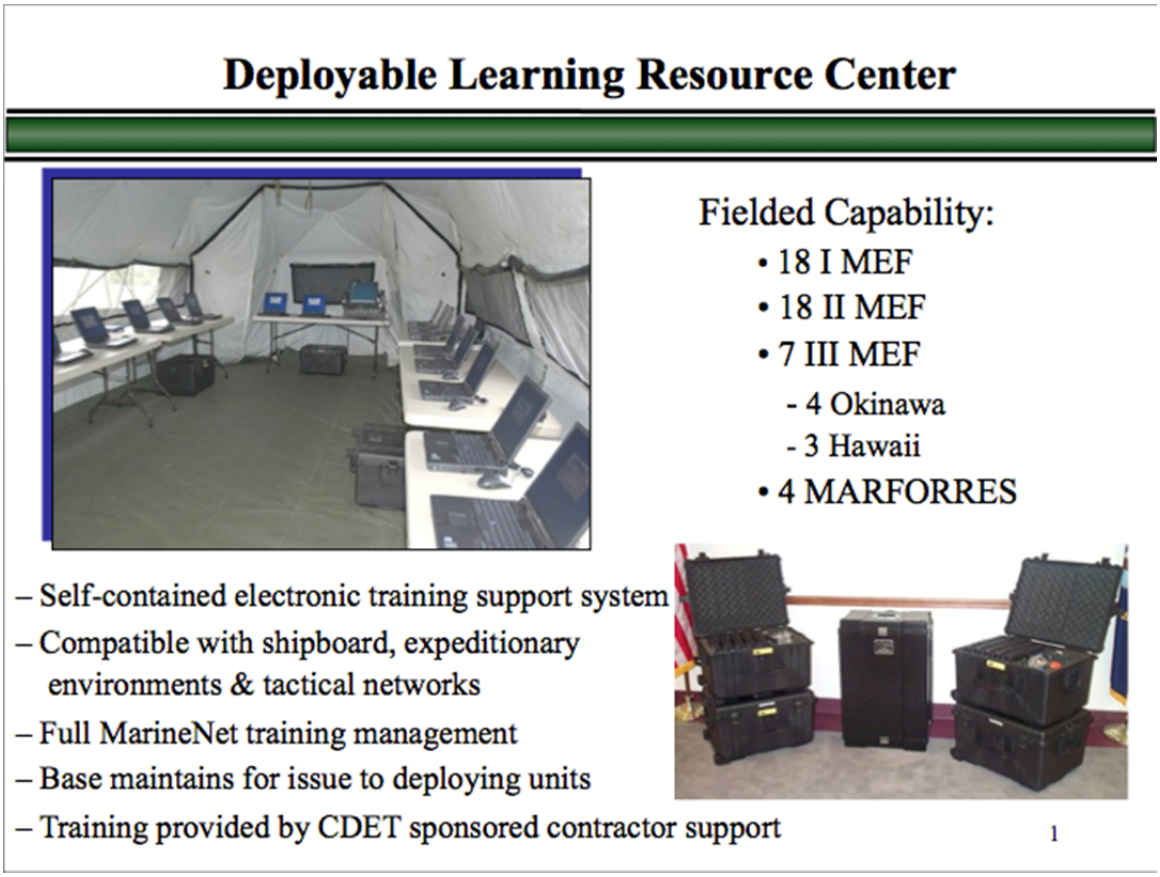


Figure 1. DLRC Solution Implemented by CDET. Source: CDET (2016).

C. NETWORK ARCHITECTURE

NIST Special Publication 800–123, *Guide to General Security*, defines a server as "a host that provides one or more services for other hosts over a network as a primary function" (Scarfone, Jansen, & Tracy, 2008, p. 2–1). Oftentimes when the term "server" is used in conversation, the immediate thought that comes to mind is a large rack that takes up a small storage room, or a bulky transit case stuffed with multiple pieces of hardware. While these certainly qualify as servers, and are applicable in a large-scale architecture, a simple desktop computer or portable laptop can easily play the role of network server. In the case of an untethered BYOD network, this is exactly what we will use to provide the necessary eLearning content to Marines and Sailors.

On a local network, there can be multiple different servers all providing various content to those clients trying to access it. Almost every network will have a Domain

Name System (DNS) server that provides lookups when a user is navigating via a browser, a File Transfer Protocol (FTP) server that allows clients to access certain documents and files, and even an email server that allows users to communicate across the network. Another important type of server used on nearly every single network is a web server, which will be the backbone of the server-based BYOD solution.

A web server is, as the name suggests, a server that provides content to a user through the web, or a browser. When a user goes into their browser and enters the uniform resource locator (URL) for the Naval Postgraduate School (www.nps.edu) in the address bar, content for that webpage is fetched from a server and provided to the requesting client. Web servers range in scale from small standalone laptops, like the one we are using for our proof of concept, to massive warehouses filled with racks in order to provide content to users across the world. For the majority of the time, web servers are hosted outside of a local network, and require tethered, Internet access in order to request and receive the content. However, if all of the content is stored locally on the machine acting as a web server, then the content can easily be accessed while disconnected from the Internet as long as the clients are appropriately connected to the web server. This is precisely what we will use when illustrating the proposed solution.

A standalone wireless laptop, which contains a pre-crafted course and a LRS, will be configured to host a web server. It will provide access to the course as well as administrator access to the LRS. The LRS, which are used in almost all eLearning environments, is the repository, or database, to which JavaScript Object Notation (JSON) statements are sent when a user takes a course. The LRS receives these JSON statements and tracks how a user performed on the course, whether they passed or failed, and when they completed the course. JSON statements can be tailored to be more informative or less informative, but they ultimately provide information about the user with relation to the course.

A wireless network, using a COTS Linksys WRT 1900 AC wireless router, will be set up and configured using Wireless Application Protocol (WAP) 2.0 encryption protocol for user authentication. A user will login to the network with the pre-shared key, and, once connected, will be dynamically given an IP address in the prescribed subnet

(192.168.1.1/24). Based on the subnet provided by the router, there are only 253 available IP addresses to be dynamically allocated to attached clients. However, because of performance degradation, we cannot expect there to be 253 clients attached at one given time We will later try to speculate roughly how many attached clients this particular router could comfortably support.

Once connected, the user will then be able to access the course through their respective browser by inputting the corresponding address. For ease of operation, our standalone server was setup as an Ubuntu machine, so the web server being employed is Apache2, a common web server used on Linux operating systems. Once the user completes the course, the administrator can login to the LRS, also by way of the Apache2 server, and see the generated JSON statements and whether or not the individual passed the course. This is all accomplished on an isolated wireless LAN that provides access to any authorized user at any given time while still protecting precious bandwidth. Any type of device, to include smart phones, tablets, and laptops, can access the material as long as they have a functioning web browser. Figure 2 displays an architecture diagram that illustrates the network used to conduct the proof of concept:

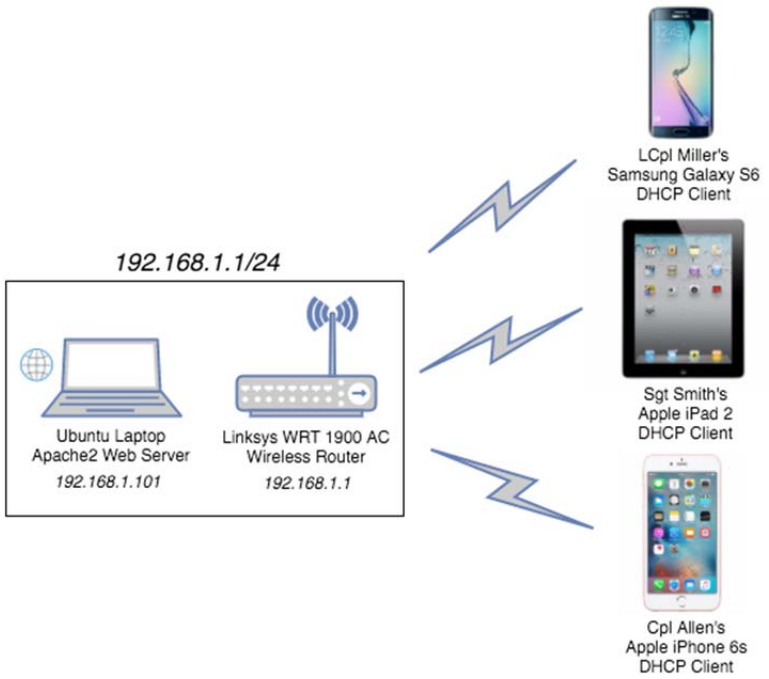


Figure 2. Server-Based BYOD Network Architecture

D. DESIGN CHARACTERISTICS

From a scalability perspective, the server-based network architecture must be designed to support a large platoon-sized element, or approximately 30 to 40 users. Later sections will explore and address the load balancing and file size considerations, to include both the courses on the server as well as the LRS's JSON storage capacity, but the proof of concept demonstrated in this thesis will be of a much smaller scale.

The two primary pieces of network equipment to be tailored and developed are the standalone web server (laptop) and the wireless router. The proposed design characteristics outlined in this section are merely to help shape future development or research, but are in no way tied to any steadfast policies, standards, or requirements. Future development of this proposed solution will require detailed policies and strategies on the approved hardware, software, and applications that can be used in a DOD BYOD environment. Currently, as was stated and amplified in the second chapter of this thesis, there simply are no guidelines, baselines, or policies in place for the introduction of BYOD into the DOD and DON. The characteristics set forth in this section were ones that led to a successful implementation of a user completing a sample course on an untethered, isolated network using their personal mobile device. The software used to host the web server, course, and LRS were all open source and free of charge.

The standalone laptop, serving as the harboring agent for the web server hosting the course and LRS, was a mildly antiquated Dell Latitude E6500. Hosting only one course and maintaining one LRS, this laptop proved to be just fine. For future, larger network environments, a much more capable laptop or robust server should be implemented. From a pure processing standpoint, this laptop would be unable to serve the needs of a small platoon because the amount of traffic and requests would simply bombard it at any given point in time. From the perspective of hosting several, most likely 15 to 20, multi-tiered courses, a fast central processing unit (CPU) is imperative to optimal network performance.

The Dell laptop used for this proof of concept, as do most modern laptops to date, possesses a 2.4-gigahertz (GHz) dual-core processor. While a dual-core processor is

extremely capable in its own right, it would not be able to service 20 service members trying to access 15 to 20 different multi-tiered courses. It is not so much the clock rate, but the amount of cores a processor contains that makes it so much more capable. A lot of modern-day robust, dedicated servers possess four, ten, or even fifteen cores. Therefore, our recommendation is for either a standalone laptop that possesses at minimum a four-core processor, or a larger, dedicated bulk server that is equipped with a ten-core, or larger, processor.

Also, the resident storage space on the server needs to be adequate to not only store the requisite, desired courses, but to also store the countless JSON messages that will be archived as members take the courses. The size of the brief, three-question sample course used in our proof of concept was 11.6 megabytes (MB). A robust, in-depth course with several graphics and interactive scenarios could be several gigabytes (GB) large, and most likely more. Typically, the size of a xAPI JSON statement is between two and four kilobytes (KB). While this is rather small, consideration must be taken into account for the multiple JSON statements generated per user for every course. If a single course generates 15 JSON statements per user, and there are 20 courses loaded on the server, then, presumably, each user would require 300 KB of space on the LRS. Although the footprint is not large, considerations must be taken into account to ensure all statements are stored properly with plenty of hard drive space.

Comfortably housing the number of courses on the server, while still allowing for plenty of space to locally record all JSON messages on the LRS, means that the server's storage capacity needs to be adequate. A server with storage of at minimum 500 GB is necessary, but 750 GB or even 1 terabyte (TB) is preferred to provide added flexibility. Varying sizes of the server could be deployed based on the amount of courses pre-loaded and the amount of users who will be accessing the material, so no one server setup will ever be fully inclusive to all situations and environments. Simply put, the server can be tailored to meet certain mission requirements, but the server, specifically the dual-core processor and amount of hard-drive storage, used in this proof of concept would not be able to serve a platoon-sized element. Figures 3 and 4 show the two devices that were used to set up the server-based network in this thesis' proof of concept:

Figure 3. Dell Latitude E6500. Source: Hinum (2012).

Figure 4. Linksys WRT 1900 AC. Source: Linksys (2016).

E. LEARNING ENVIRONMENT AND NETWORK TRAFFIC

Now that the critical pieces of hardware for the network have been identified, we will explain how we set up our server-based learning environment and how a mobile device connects to the network and accesses the course. We will then give a cursory glance into the traffic that is occurring between device, server, and router.

1. Environment Setup

After the Apache2 web server is installed and set up on the Ubuntu Dell Latitude, the next task at hand is to create and install the LRS to store JSON statements from the course in order to track a user's progress and performance. Most large-scale record stores that are used by organizations are serviced and maintained by the individuals who created them, usually for a nominal fee or subscription. For the case of our proof of concept, we needed to implement a small-scale, open-source LRS that could retain statements from a small, sample course. We ultimately decided to use Learning Locker, which is an open-source, multi-platform LRS that is xAPI compliant. The main reason why we chose a Linux operating system for our server was because installation and setup of Learning Locker, which the Apache2 web server hosts, with Ubuntu was substantially easier than Windows or Mac. The course we created was a simple three-question quiz that any Marine should master. The idea here was not to create a robust course to test the learning environment, but to more importantly show feasibility of the network we are trying to construct. It goes without saying that most courses hosted by the DOD and DON will be multi-tiered, have clearly crafted learning objectives, and will be much more robust in nature. For simplicity, we used an open source website (www.easygenerator.com) to create our course rather than build a full-fledged html file from the ground up. Again, the idea was to implement our proof of concept and not try to replicate a DOD or DON course. An added benefit though is that all courses constructed using www.easygenerator.com implement the cutting-edge xAPI standard, which the Marine Corps is looking to implement in the not too distant future.

Setup of the LRS proved to be somewhat challenging due to the fact that almost all record stores are never locally hosted, nor are they ever really developed by the user. Most record stores are cloud-based, requiring Internet connectivity, and are developed and tailored for the customer, in this case the DOD and DON, by the administrators. Because of that, we had to rely on a lot of personal experiences, mostly via forums and blogs, and documented user trials in order to properly setup our untethered local LRS. After installation, part of the setup for the LRS allowed for one or more administrators to be created, preventing any individual with network connectivity to access the records of

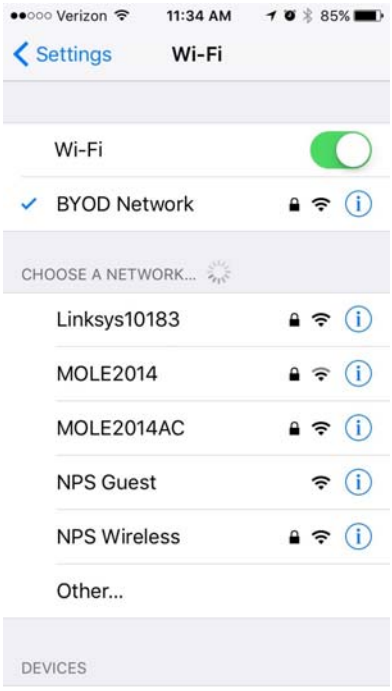
the LRS. To access the LRS and see its contents, it required a valid username and password upon lookup. This proves to be extremely necessary since an individual can pull up the web address of the LRS because it is hosted by the Apache2 server.

Once the LRS was set up, we needed to ensure that JSON statements could actually be sent to the endpoint and logged locally. The endpoint of an LRS is, as the name suggests, the location where the course sends all statements for record keeping. For cloud-based LRS setups, this is typically a URL to a server where they can manage thousands of different record stores. Any course built using Easy Generator would have a default endpoint to an LRS server that was managed and maintained by the Easy Generator administrators. For our network to work and truly be untethered, we needed to adjust the endpoint so that all JSON statements would be sent to our local LRS and not the cloud-based Easy Generator LRS. The endpoint was going to be the private Internet Protocol (IP) address for the LRS, which we were able to determine upon setup of the LRS. For our network, the endpoint was http://192.168.1.101:663/data/xAPI/. As was mentioned before, our IP range for the private network provided by the Linksys router is 192.168.1.1/24. We gave the server a static IP address (192.168.1.101) because configurations for the course and endpoint were already made in previous experiments with a less capable router that dynamically gave the server that particular address. For ease, and so we did not have to re-create the course with a new endpoint using the IP that the new Linksys router provided, we simply assigned the server the same static IP address that fell within the Linksys router's subnet.

Now that the LRS was set up and our course had the correct endpoint established for record keeping, we needed to find a way to enable the users to access the course via the browser on their personal device. To do this, we created two virtual hosts, one for the LRS and one for the course. Both hosts would use the same IP address, but would be listening for queries on different ports. Arbitrarily, we selected the LRS to listen on port 663, while the course would listen on port 8080. So, when an individual input the address 192.168.1.101:8080 in the browser's address bar, the course would effectively launch. If an administrator wanted to access the LRS from their personal device, they would simply type 192.168.1.101:663, and the LRS login page would appear. This was all capable

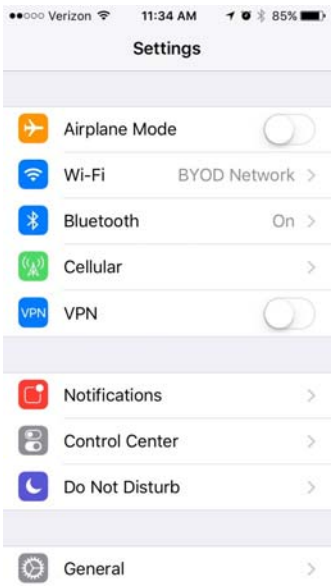
because of the Apache2 server. We could have created a DNS server to allow users to input a URL for ease of lookup, but decided against it since this thesis is not concerned with the detailed features of the network, but more the overall successful application. A more refined, final network setup would include a robust DNS server that would allow a Marine or Sailor to simply type *MarineCorpsCourse.com* and have the course launch without having the remember the IP address and port number.

Our network environment is now setup and fully operational. A user has the ability to log on to the private, untethered network (aptly named "BYOD Network") that is hosted by the Linksys router, they are able to launch the three-question Marine Corps course, take the course, and the LRS receives the JSON statements locally and tracks the user's progress. Figures 5 through 12 are screenshots of the step-by-step process a user encountered on their personal iPhone when taking the course, as well as screenshots of the LRS receiving the JSON statements.



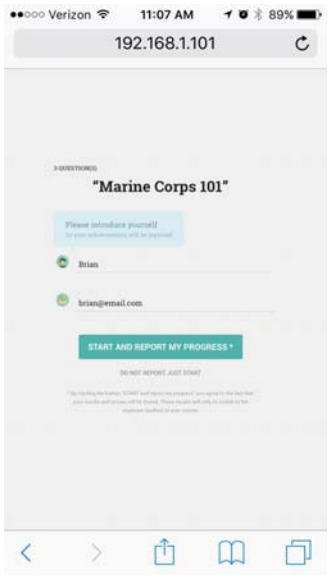
The screen of a user connecting to the private, untethered BYOD Network using their personal iPhone 5S. The network is password enabled and encrypted using WPA2.

Figure 5. Personal Mobile Device Wireless Networks Screen



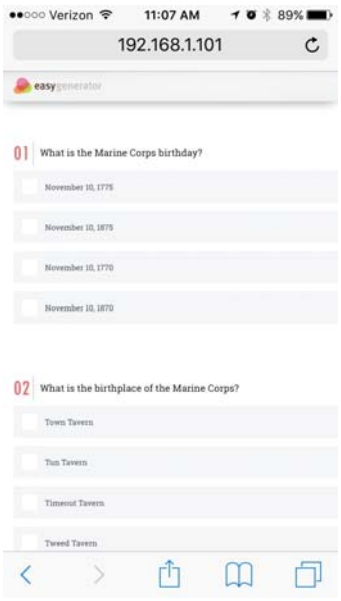
The user has successfully connected to the BYOD network using the pre-shared key. Only users who require access to the network will be provided the pre-shared key for network access.

Figure 6. Personal Mobile Device Network Connection Screen



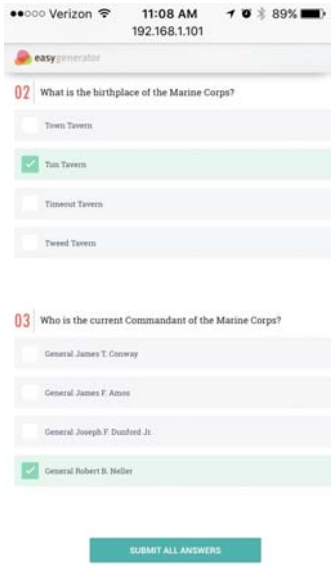
The course is launched after the user types “192.168.1.101:8080” into their browser’s address bar. A login screen is the first screen the user will see. They will provide their name and an email address in order for the LRS to track their progress. More robust iterations of the LRS will have pre-defined accounts and usernames that will track a multiple users’ progress over time, but this LRS does not provide that functionality.

Figure 7. Course Launch and Login Screen



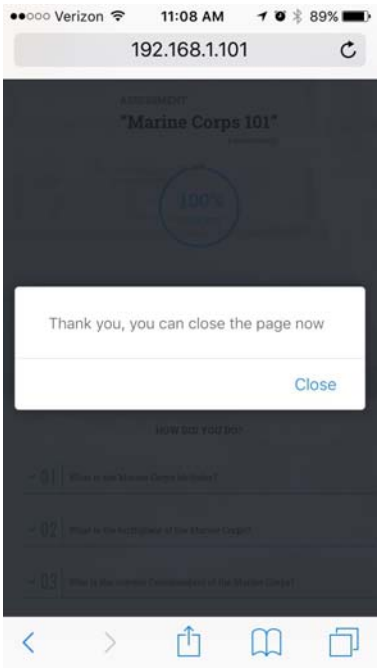
A screenshot of the three-question sample course that was created for the proof of concept. The user selects the answer they feel is correct, and submits all three answers at the bottom of the page.

Figure 8. Sample Course Screenshot



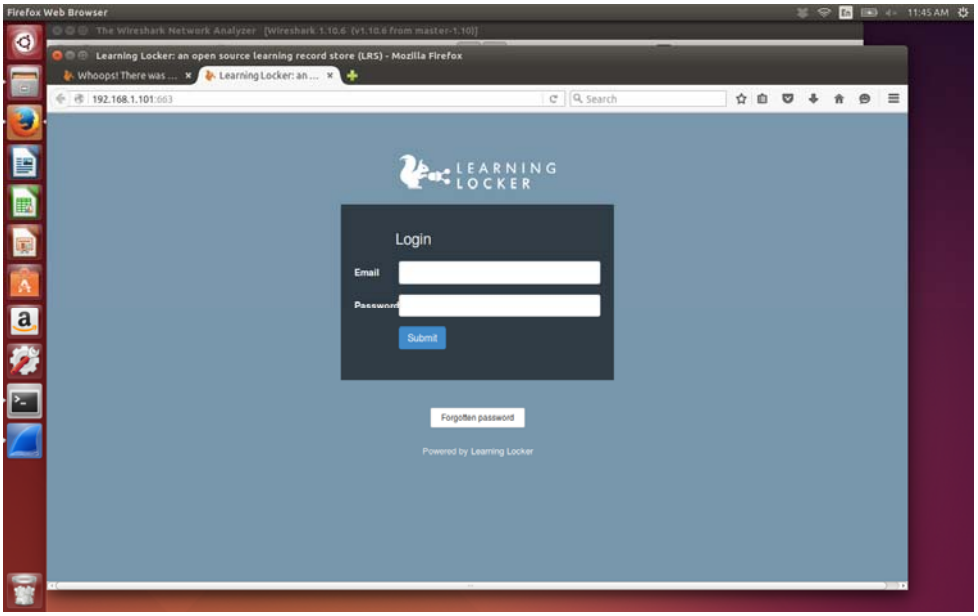
The user selects the answers and presses the “Submit All Answers” button shown at the bottom of the page. Once this button is pressed, all results will be sent to the LRS for record keeping.

Figure 9. Sample Course Selected Answers Screenshot



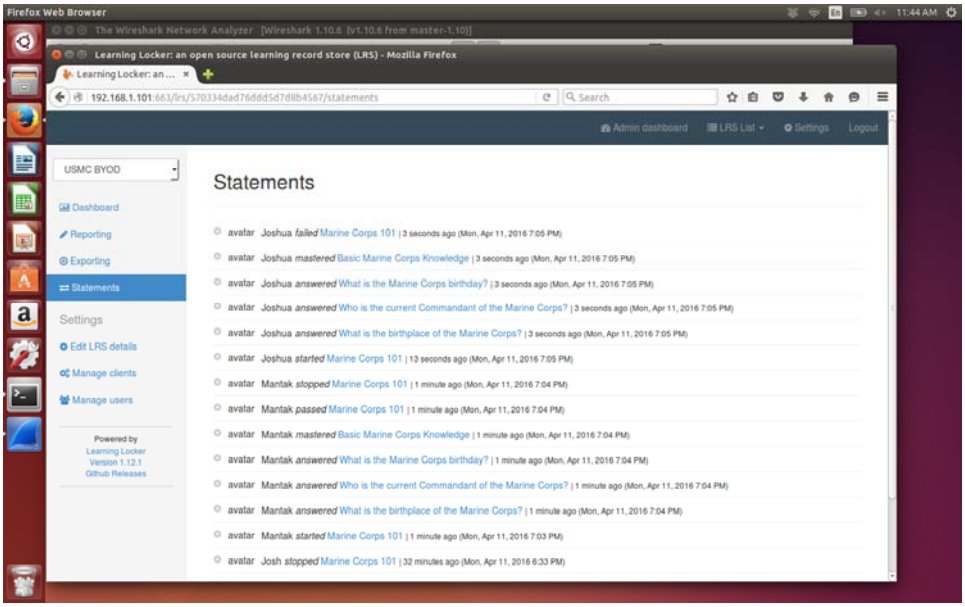
Once the answers are submitted, a page is loaded that shows the results and how the user performed. Then, the above window is displayed when the user clicks “Finish Course.” The user has now completed the course after having joined the private network using their personal mobile device.

Figure 10. Course Completion Screenshot



Once the course has been completed, an administrator can login to the LRS and see the progress of the users. We are able to navigate to the LRS by typing 192.168.1.101:663. This is what appears when the browser does the lookup.

Figure 11. LRS Login Page



Here we can navigate to the LRS "USMC BYOD" and see all of the statements that have been sent when users take the course. A statement is sent when each user starts the course, stops the course, answers a question, passes, or fails the course. All of these statements receive a timestamp, and can be graphically represented using the built-in Learning Locker features.

Figure 12. LRS JSON Statements Page

2. Network Traffic

Because of the untethered nature of the BYOD network we set up, there is not a lot of complexity surrounding the traffic it generates. Due to the fact that all course files are saved locally on the server, the only traffic that is generated upon a user connecting to the network comes in the form of a simple DNS lookup. After running both Wireshark and Debookee, which are two packet capture applications that aid in traffic analysis, we are able to see that, upon inputting the 192.168.1.101:8080 IP address to launch the course, a simple *fonts.googleapis.com* lookup is performed and the course is launched. Other standard Address Resolution Protocol (ARP) traffic continues to flood the network, but the only legitimate traffic that the personal mobile device generates is the DNS lookup upon entering the IP into the browser. The course is saved locally, so all that occurs is a lookup to the index.html file for the course associated with the given IP address. Once the course is launched, there is no content fetching required because the course is hosted locally on the machine. This would be the same for any number of courses the BYOD network could host via the Apache2 web server.

F. LOAD BALANCING AND NETWORK CAPACITY

Because the proposed BYOD network is rather small in nature, the load balancing aspects of its functionality rely solely on the router and access point. The server's computational power, per the somewhat speculative design characteristics and performance features stated previously, should more than suffice for a platoon-sized element of about 30 to 40 Marines or Sailors. However, it is the sudden influx of users joining and retrieving material on the network where service members may experience some connectivity or content delivery issues.

Although the private 192.168.1.1/24 network can adequately provide 253 unique host addresses, it is not feasible for the router and server to be able to handle this amount of traffic, especially at one given time. Like all routers do, they would perform dynamic load balancing based on signal strength in order to help those clients who are attached with stronger signals to receive the content they are requesting. When a client's signal strength degrades and the server struggles to provide that material real-time in the

network, the router would recognize this and drop that particular client to free-up precious bandwidth for other users who have a more established, reliable Wi-Fi connection.

In order to effectively evaluate the true network capacity of the router and server hosting the material, the network would need to be setup with legitimate, accredited MarineNet and NKO courses that contain far more graphical content and computational processing than our proof of concept. However, from a pure speculative standpoint, it is our professional opinion that ten to twelve users could comfortably access material at a given time without interrupted network connectivity. Again, this is largely dependent upon the amount and size of the courses the server is hosting. Due to the fact that MarineNet courses range in size from an extremely small flat file to a robust, graphics-laden bundle, it is hard to develop the mean baseline without tailoring a specific server to unit's desired course suite.

The proof of concept in this chapter is not intended to integrate BYOD into the larger DOD network enterprise, but to provide deployable, small-pocket solutions that can be implemented one hundred times over in units spread across the world. Further research would need to be conducted in order to accurately depict the network responsiveness of multiple users requesting legitimate learning content in a range of different environments. Because the research and development has not evolved to this point yet, we must purely give our best estimate as to the ideal network capacity of the proposed solution. This is an added area of research in this domain, and will be discussed in detail in our final chapter.

G. SUMMARY

Currently, MarineNet hosts over 2,250 courses ranging from operational security to range safety qualifications to even nutrition (Smith, 2015). Additionally, there are approximately 325,000 active users, predominantly active duty or reserve service members, who access this content on a daily basis (Smith, 2015). While this proposed server-based solution, in its singular form, is only intended to aid 30 to 40 users during a

given period of time, the overall intent is to field a capability that harbors enhanced user interaction while also serving as a force multiplier.

So often in the computer science industry the term “quality of service” or “QoS” is used as a linchpin for commercial research and development of retail products intended for user purchase. It is time, from an eLearning standpoint, that the DOD and DON begin thinking about improving the QoS for the users who access these various materials. It is foolish to expect thousands upon thousands of these routers and servers to be distributed and the networks be employed to accommodate every possible user. However, as much of the commercial world has begun to realize, we need to harness the computational power of a user’s mobile device. As technology evolves and tablets and smartphones become increasingly more powerful, it needs to be a strong consideration, especially for the DOD.

As the literature in Chapter II alluded to, or did not allude to based on the lack of instruction on the matter, the only way a successful BYOD environment will be implemented across the DOD is in an untethered capacity, completely devoid of risk and consternation. From a scalability perspective, this server-based solution may not be the most ideal and may not provide maximum utility across all facets of the DOD. However, it is a solution that can be tailored to a unit’s training requirements, deployed to any clime or place, and will prove cost-effective in the long-term. Overall, we feel that the server-based solution is a credible, viable starting point moving forward to facilitate the integration of personal mobile devices into the DOD, specifically the Navy and the Marine Corps.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. SERVER-LESS BYOD SOLUTION

A. MODES OF OPERATION

For the purposes of this study we have selected to use an application stored on a personally owned electronic device that will use local resources such as battery, memory, Wi-Fi, etc. The application itself needs to be operable in both online and offline modes because the necessity for using the device for eLearning aboard a military vessel exists both while at port and underway. It is intended for this application to be used for both the server and server-less approaches to this study.

1. Server-Based Operating Modes

When operating on a ship-based server, the smart device, via the operating system's browser, will connect to and access course content stored on the local server as well as return course completion statistics in a JSON message. The server will then be responsible for communicating with the CDET learning management system to update student portfolios and transcripts. This communication between the server and the learning management system can occur manually by a system administrator offloading the JSON statements, or, through further development, can be done automatically through an application interface.

2. Server-less Operating Modes

For the server-less method of this study, it is imperative that the application communicates with CDET servers via the Internet prior to entering offline mode. This will allow course content to be pre-loaded and stored locally on the device. At the conclusion of an eLearning segment, course completion statistics will be compiled and stored locally, on the personal device, until connectivity is restored while underway or at port. One of the main focus areas of the server-less approach will be sending the completion message containing course name, version, user name, score, and date via Short Message Service (SMS), also known as a text, email, or picture message. By leveraging existing infrastructure to send the message to CDET, the military eliminates

the need to procure, build, test, maintain, and replace eLearning-specific hardware on every ship. This approach also minimizes personal device computational requirements and allows CDET servers to perform the bulk of the necessary processing. Utilizing this cloud-type approach will save bandwidth, battery, and money.

B. USER INTERFACE

For a Marine to successfully complete a required course from a personal mobile device, an acceptable, well-organized and simple user interface is necessary.

1. Online Functionality

The online mode of the application will be used for searching the CDET course catalog for one of the approximately 2,250 courses currently hosted on MarineNet (Smith, 2015). Once a course has been identified and selected, the online mode allows the Marine to read a more specific course description, enroll in the desired course, then download and save it locally to their personal device. As the Marine is working on the selected course, no online functionality is required until the course has been completed. Upon completion, online functionality will be used to email either a JSON message or hash code developed by the application directly to CDET's learning management system. All other functions operate in offline mode and do not require Internet connectivity. See red arrows in Figure 13 to identify required online functions.

2. Offline Functionality

The main purpose of the server-less approach to eLearning is to be able to work on and complete courses and learning objectives remotely while offline and disconnected from a server. The application's user interface will allow a student to see which courses are currently stored on their device, open and launch a class, track course progress for segmented completion, view transcript of enrolled and completed courses, view grades received from completed courses and set up a transmission to CDET to report course completion, all while operating without Internet connectivity. The bulk of the user interface will operate internally with the only external communication requirement being course download and completion message upload.

3. Connecting to Local Server

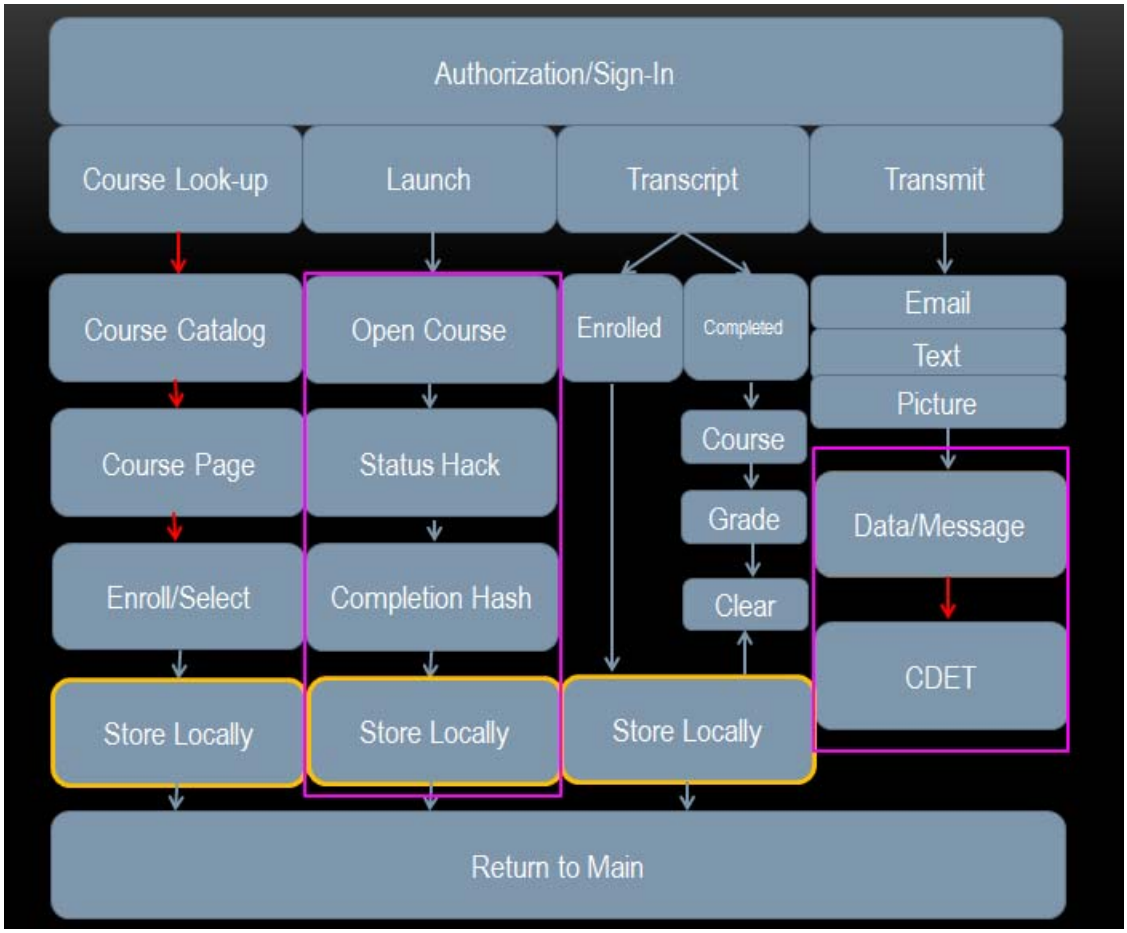
There will be times when Internet connectivity is non-existent. During times like this it will be important for the user interface to be able to connect to local infrastructure such as a ship-based server. This option will be available for both the server-less and server-based solutions because being able to connect to local infrastructure could be beneficial for storage, downloads, and developing an ad hoc network.

4. Resources Used on Device

Due to the heterogeneity of personal devices, it is difficult to say exactly how all the resources will be used on a given device by the user interface. Personally owned electronic devices have a wide range of operating systems, updates, architecture, memory capability, screen sizes, and battery life. A personal computer (PC) will have a higher capability to store courses than a smart phone. The PC may be able to store a year's worth of annual training locally whereas a smart phone may only be able to store two segments of one course. These discrepancies will need to be dealt with internally as the application is loaded onto a device. A settings option can help determine how much memory is allocated for the eLearning courses and transcript storage. By adjusting settings, a user is able to determine how much impact the eLearning application has on the overall functionality of their personal device.

5. High Level Overview of Application

After successful authorization and log in, the user interface will have four main functions on the home screen. These provide the student the means to look up a course, launch a course, view transcript, and transmit. Within each main function, a student can perform an important element of course completion. All four processes link back to the home screen and can access local memory. Figure 13 illustrates the functionality and interaction of the application with the online and offline components.



Red arrows depict online requirements. Blue arrows depict offline capable functions. The purple rectangles show the proof of concept portion of this study.

Figure 13. High-Level Diagram Depicting Application Functionality

C. AUTHORIZATION / SIGN IN

To log into the application and use the functionality within, it is mandatory to strictly control who has access to DOD course material and the CDET servers. This process can be done using a variety of access control methods. Access control encompasses four key factors: identification, authentication, authorization, and access (DISA, n.d.).

1. Identification

Identification is a physical control process that uses a person's identity when requesting access. In order to gain access to a system in the DOD, an individual must first go through a credentialing process. Credentials are granted based upon a user's required level of access and mission essential need to access a given system. This prevents unnecessary access and limits the risk a system will be exposed to. Those that do not require access are simply not credentialed to access it. The DOD requires a four-step credentialing process (DISA, n.d.):

- Validate the identity of requester
- Identity registration
- Credential generation through PKI or other means
- Combined the identity to an authentication method

Once the credential has been created, validated and registered, a user has been given permission to access a DOD system.

2. Authentication

Once an individual has been credentialed, they must prove their identity every time they attempt to log on to a designated system. This process is known as authentication. Authentication is a process that uses a validating credential to prove a given identity. This can be done in three ways: knowledge-based, token-based, or biometric-based. Depending on the system being accessed and the material it stores, the DOD can use any one, combination of two, or all three of these authentication methods for every login. For sensitive material, a three-factor authentication minimizes the risk of an unauthorized user gaining access to a system. Tables 2 through 5 provide examples of the different single and multiple factor authentication methods.

Table 2. Examples of Token-Based Authentication.
Adapted from Mitchell (2014).

Method	Description
Decal	Decal mounted on a motorized vehicle.
Transponder	Transponder used for an automated entry point.
Badge	Not personalized (e.g., visit badge without name/photo).
Key	Physical key of any kind.
Memory Card	Memory cards without the PIN, whether personalized or not (e.g., magnetic strip, barcode, optical or smart cards used as memory cards).
Smart Card	Personalized or not. Includes cryptographic and non-cryptographic cards. Includes all communications interface types (e.g., contact, contactless).
Digital Signature	Issued by DOD-approved PKI authority.

Table 3. Examples of Knowledge-Based Authentication.
Adapted from Mitchell (2014).

Method	Description
Password	DOD compliant password or PIN.
Unshared Combination	Electronic safe, cipher lock, or PIN pad combination which allows individualized PINs or combinations.
Shared Combination	Safe, cipher lock, or PIN pad combination with shared combination.

Table 4. Examples of Biometric-Based Authentication.
Adapted from Mitchell (2014).

Method	Description
Colleague Recognition	Personal recognition by peers and co-workers. Considered to be attended access. Document policy and train users.
User Recognition	Attended access control implementations wherein peers or security guard/personnel perform identification and authentication. Document policy and train users.
Fingerprint Identification	Fingerprint authentication using one-to-many match against templates or images stored in a remote database. This does not match a fingerprint found on a card.
Fingerprint Verification	Fingerprint authentication using one-to-one match against templates or images stored on the CAC biometric reference database.
Hand Geometry	Hand Geometry authentication using one-to-many match against templates or images of various characteristics of the hand and finger measurements (not fingerprints) stored in a remote database.
Iris Scan	Iris Scan authentication using one-to-many match against templates or images of the eye stored in a remote database.
Voice Recognition	Vocal scan compared to database.

Table 5. Examples of Multiple Factor Authentication. Adapted from Mitchell (2014).

Method	Factors	Description
Cryptographic Hardware Token	Token Knowledge	FIPS 140–2 or NSA certified encryption module used in cryptographic hardware token to implement one time password device and PIN or password solution.
Photo ID	Token Biometric	Verified digital or optical photo ID. Use of approved procedures for verifying a non-CAC photo identification card (e.g., driver's license)
Spoken Password	Token Biometric	Key phrase only the user knows is spoken and analyzed. Scan is compared to database.
CAC Photo	Token Biometric	Procedure for verifying the photo on the CAC.
CAC	Token	Implies that its presence and validity is verified by an automated system (e.g., a swipe into a reader). The purpose is to validate that it is an authentic CAC card.
	Token Knowledge	CAC with PIN for after-hours entry into vacant workspace without after-hours attendant.
	Token Knowledge Biometric	Attended or two-person access control using a CAC plus PIN.

3. Authorization

Authorization is the final step in granting access to a system. By proving who you are through identification and authentication, your requested access is compared to the access registered to your identity. This is referred to as access control. Ensuring individuals receive only the access they require, helps ensure the integrity and security of the DOD system.

4. eLearning Application Access Control

For the eLearning application on a personal electronic device it is important to select an access control method that is feasible, simple and not cost prohibitive. As technology continues to advance beyond usernames and passwords, it is important the DOD continues to evolve their policy toward biometrics. PEDs today now have the ability to unlock based on thumbprints, pattern input recognition, and facial recognition. The new Windows 10 software offers "Windows Hello" which uses facial recognition to log in by using the onboard front facing camera to verify the user's identity (Microsoft,

n.d). Microsoft expects this to become a standard log in procedure in the future and can be combined with fingerprint recognition to create an even more secure way to get instant access to your computer. We can expect this technology to make its way to smart phones in the very near future and should prepare future DOD products to accommodate these technology advances.

Using a token-based authentication for PEDs is impractical. Young Marines on a ship cannot be expected to purchase their own smart card reader in order to conduct online training from their own devices. This puts an unnecessary financial burden either on them or the command supplying these smart card readers. Using a smart card to log in remains one of the most secure methods for authentication as distribution control is tightly monitored, but this token-based access will not allow the eLearning application to reach its full potential. The hardware cost is just too high for eLearning purposes.

One potential means to overcome the financial burden of smart cards being used is to develop a method to store smart card credentials on a device. Requesting programs can access this stored certificate and prompt the users to provide their standard smart card pin to activate the certificate. This would eliminate the need for portable smart card readers, while at the same time continue the token-based authentication the DOD has grown accustom to. The downside to this method is the security associated with storing the certificate electronically.

Today, MarineNet users continue to access CDET systems through a username and password protocol, bypassing the requirement to use a smart card. This authority to operate was granted in 2015 when the system was approved as is by Dr. Raymond Letteer, Chief of Cyber Security Division, United States Marine Corps. For the eLearning application we are presenting, it will be necessary to ensure the smart card waiver is extended to all BYOD applications. The courses listed on MarineNet are not classified and therefore prove little risk to the government. As for now, the best and most practical approach for access control will remain username and password, but attention should be shifted toward biometric options in the future.

D. COURSE LOOK-UP

A key aspect for the server-less method working for this eLearning application is to ensure Marines can select and download required courses. With over 2,250 courses on MarineNet, it is important to provide a quick and effective search method to find and enroll in the correct courses (Smith, 2015).

This functionality of the application must be performed while connected to the Internet. The application will connect to the CDET course registry and display available course content. Some courses may not be accessible from a mobile device. If the size of the course or aspects of the course limit a student to viewing from a PC only, it is important for the user to see this restriction while reading a course description. In fiscal year 2015 alone, MarineNet serviced over 5.62 million course enrollments. Of these enrollments, MarineNet registered 3.99 million course completions (Smith L. E., 2015). It is imperative that students looking to register for a course can find the correct course and version they have been asked to complete. Within the course look-up section of the application, a Marine will have the ability to search available courses, read course descriptions and inspect curriculum requirements for a pre-planned program. By dividing the course look-up into annual training, Marine Corps Instructions, and miscellaneous courses, a Marine can easily identify and enroll in necessary courses.

Once a Marine has identified a course they would either like to take or have to take, they simply click the enroll/download button and the course is automatically stored locally on their smart phone, or other electronic device. A key feature that may be required is a size filter for mobile phones. Due to limited memory and storage space, some courses may need to be broken up into storable segments for some users. Smart phones can range in storage from less than 4 GB to more than 164 GB. That is a wide range of available storage and the eLearning application must adopt a method to satisfy users from both ends of the storage spectrum. If a user has intentionally limited the eLearning application to using 2 GB of local storage, and the course they are would like to complete is 4 GB in size, it is necessary for the course to be completed in segments. If the course is an exam, the results from each segment must be stored locally until the course is completed and a final score is tallied. This segmented approach will prove to be

valuable, as it will allow more users to conduct required training from their personal device. The overall goal of this project is to make eLearning more accessible for Marines as they perform their duties around the globe.

E. LAUNCH

To begin a course that has been downloaded and stored on a personal electronic device, the user clicks the "launch" button from the main screen. This opens up a new page that searches the memory of the device and lists the stored courses with completion status, and allows the user to select the desired course. Once a course has been selected, course specific instructions will be displayed and a final "begin" option will launch the course and start the timer if the course requires a timed session. If the user is returning to complete a previously started course, it will begin at the last recorded or saved spot.

As all the "launch" functionality will be conducted offline, it is important for the application to track the progress of the course. Some classes may require start-to-finish completion. In these cases, the application will inform the user that saving work is not available. When a course is completed successfully on a device, the application will need to generate a completion message that will be sent to CDET for tracking purposes in their learning management system.

F. COMPLETION MESSAGE

When a course is completed successfully on a personal electronic device, the application will need to generate a completion message that will be sent to CDET for tracking purposes in their learning management system. This is the key functionality of the application that sets it apart from the normal online registry.

At the completion of a normal online MarineNet course the system automatically generates a JSON message that sends pertinent information to the CDET learning management system. This message includes the course name, version number, date and time the course was taken, which registered user completed the course, passing grade of the course, and other important criteria that CDET requests. The JSON message is designed in a way to allow automatic updating of a user's profile and transcript. The

eLearning application we are proposing will have similar functionality when connected to the Internet, but will also feature a new and unique way to transmit a completion message when not connected to the Internet.

1. Completion Message Recording Concept

In order to successfully use a personal electronic device for eLearning without being connected to the Internet, a user must be able to communicate a completion message to the CDET Learning Management System. The message must be short, easy to transfer over existing cellular infrastructure and difficult to replicate for co-located users. To solve this problem, we are proposing a hash value based system that would require minimal adaptation to the current scheme that is transmitted via the Short Message Service function of a user's mobile smart phone.

Before exploring the details of the hash system, it is important to provide an overview of the concept. When a student registers for a course and downloads it onto his or her smart phone, the CDET LRS will automatically log specific information into a database. The student's name, ID, email address, course name and number, course version, and download date will be provided and then stored both locally in the LRS and also be imbedded in the course when the user downloads it. It is these pieces of information that will be used in order to generate a unique hash in the CDET LRS and also on the user's personal device upon course completion. Along with the basic profile information of the student and course, the database will store a set of hash values corresponding to the downloaded course. These hash values are computed using the user provided information upon course registration and download. For simplicity's sake we will consider a pass/fail course to limit the number of hash values that need to be stored. Figure 14 depicts the computation performed by CDET to generate two distinct hash values for a set of information. These hash values are stored in the database alongside the information previously mentioned. The user's information that was provided during registration is imbedded in the course upon download to be used for pass/fail hash generation once the course is finished. When the user completes this course, this information is passed to the mobile application with the appropriate "pass" or "fail"

status appended where the application's resident hash function produces the completion hash. This hash digest can then be sent via SMS from the student to the CDET LRS. When the LRS receives the hash, it compares it to its database of pre-generated and stored values. If the LRS finds a match, it can then determine what course is being reported as completed, by which student and in turn populate the transcript with the pertinent information and update the LRS.

a. SMS Text Messaging

According to the Pew Research Center, text messaging is the most widely and frequently used application on a smartphone, with nearly 97% of Americans using the functionality on a daily basis (Smith A., 2015). Due to the largely expanded use of text messaging from areas such as marketing, social interaction, and information dissemination, it is important to explore alternate methods to exploit the benefits of text message proliferation for all areas of life.

Text messages are limited in character length to 140 octets or 1120 bits by the signaling protocol. This translates to precisely 140 8-bit characters. A typical JSON message will contain more information than is available to be sent via SMS. This demonstrates the need to strip a JSON message of its critical reporting criteria and reformat the information to make it transmittable via the limited 140-character set on a text message. By extracting the course code, student identifier, date, score, and total time a course took to complete from a completed course's statistics we can repackage this data into a hash value that is less than 140 characters, negating the limitation presented by SMS.

b. Hash Function

A hash function is used in many different areas of cryptography and computer science. Over the years, different hash functions have been developed and broken as technology continues to advance and computing power grows. A basic hash function takes an arbitrary sized input and creates a fixed size output that is relatively easy to compute yet very difficult to reproduce and only vulnerable to a brute force attack. This trait is referred to as non-invertible or one-way function. When given the hash value $h(x)$,

it would be next to impossible to reconstruct the original message data *h* without computing all possible values searching for a match.

An ideal cryptographic hash function will create a hash digest that is unique to one specific data input. It should be nearly impossible to find another message that produces the same unique hash digest (Gondree, 2014). This is known as collision resistance; two inputs a and b such that $H(a) = H(b)$, where $a \neq b$. It is important to select a hash function that limits collisions. In our case, the automated learning management system needs to successfully populate the correct student's record with accurate scores for the proper classes taken. Collisions would result in the learning management system being unable to decipher who the received hash digest belongs to and what scores need to be transfer to a transcript.

Figure 14 depicts four pieces of input information commonly found in a JSON message communicated to a learning management system. This data was hashed using an MD5 hash function to produce a 32-digit alphanumeric code commonly called a hash digest. To show the uniqueness property the 'score' data was changed from 100 to 72 to demonstrate the affect a small change can have on a hash digest.

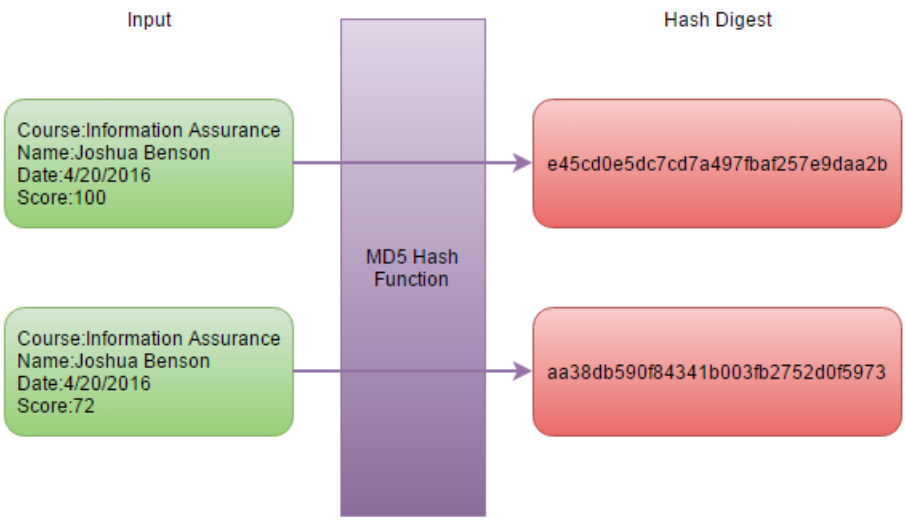


Figure 14. Inputs Hashed With MD5 Function Producing Hash Digest.

Another important feature a good cryptographic hash function will provide is the avalanche effect. This occurs when one small change in data input, for instance changing a lowercase “a” to an uppercase “A,” creates a drastic change in the hash digest output. This can be seen in Figure 15. Using a SHA-1 hash function, the hash value is completely different for the two inputs, even though the only change was the capitalization of one character.

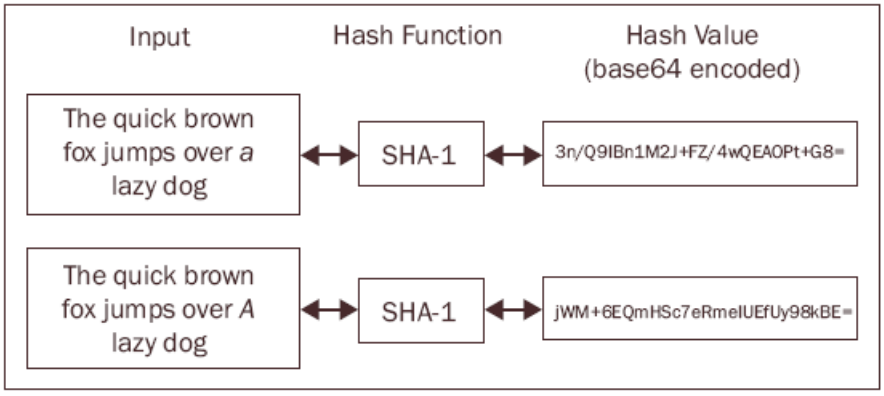


Figure 15. Avalanche Effect. Source: Paradigm (2008).

In cryptography, hash functions are used to preserve the integrity of stored data. If a hash digest does not match the original data’s hash digest, then it can be determined that the newly received data has been modified by an external source. This does not provide authentication however. Adding authentication would require the use of a keyed-hash message authentication code (HMAC). An HMAC uses a standard hash function in conjunction with a special cryptographic key. Combining these techniques produces data integrity and authentication that does not exist when used individually.

For the purposes of our proposed eLearning application, we feel that a simple hash function alone will suffice because authentication will be achieved through our sign-in and course download procedure. Unlike in cryptography where an adversary is trying to intercept and decode data, the eLearning application’s “adversary” would be multiple students attempting to forge a completion message. If they were to determine what hash function is being used to create the digest, they could in theory attempt to input

completion message criteria, such as course, name and score, through their own hash generator and submit course completion to the learning management system.

To prevent unauthorized course completion messages from populating in the learning record system, we could implement a salt into our hashed data. Salt is a unique use of random characters appended to data prior to running through a hash generator. As long as the salt is randomly chosen and unknown to the user, replicating a valid hash digest would be impossible. When the message is received by CDET and compared to a list expected incoming hash values, the salt can be removed and data processed accordingly.

Another option to prevent false reporting would be to create a unique user identification code for every course downloaded to a personal electronic device. This code would be stored in the learning management system and used in the hash function to create a hash value that CDET could decipher, yet be impossible to replicate. If another student were to attempt to submit a hash digest to CDET without the registered user code appended at a specific place in the data, the submitted hash would not match the expected value and thus be denied by the LRS and CDET would be notified of the improper attempt.

For courses that require a passing grade, the hashed message may contain permanent information such as course name, student name, unique download ID, and date, and append the user's course score at the end. This would limit the number of hashes required to be stored in the learning management system. A received hash digest itself will not be able to be decoded by CDET. CDET will only use a received hash digest to compare to their database of expected and appropriate hash values. For instance, when pass/fail course 91A is downloaded by a user named Mike. CDET will assign the value 2c3 to the download. CDET will also create two hash digests to represent the possible outcomes of Mike submitting his course grade to the LRS. Mike's pass and fail codes are then saved in a database. Once Mike completes course 91A and his phone creates the hash, he can then transmits this to CDET via text or email. CDET will compare the received hash with its database and be able to determine that because code uUjk3fR5 was

submitted, Mike successfully completed course 91A. The front-end work conducted by CDET can be seen in Figure 16.

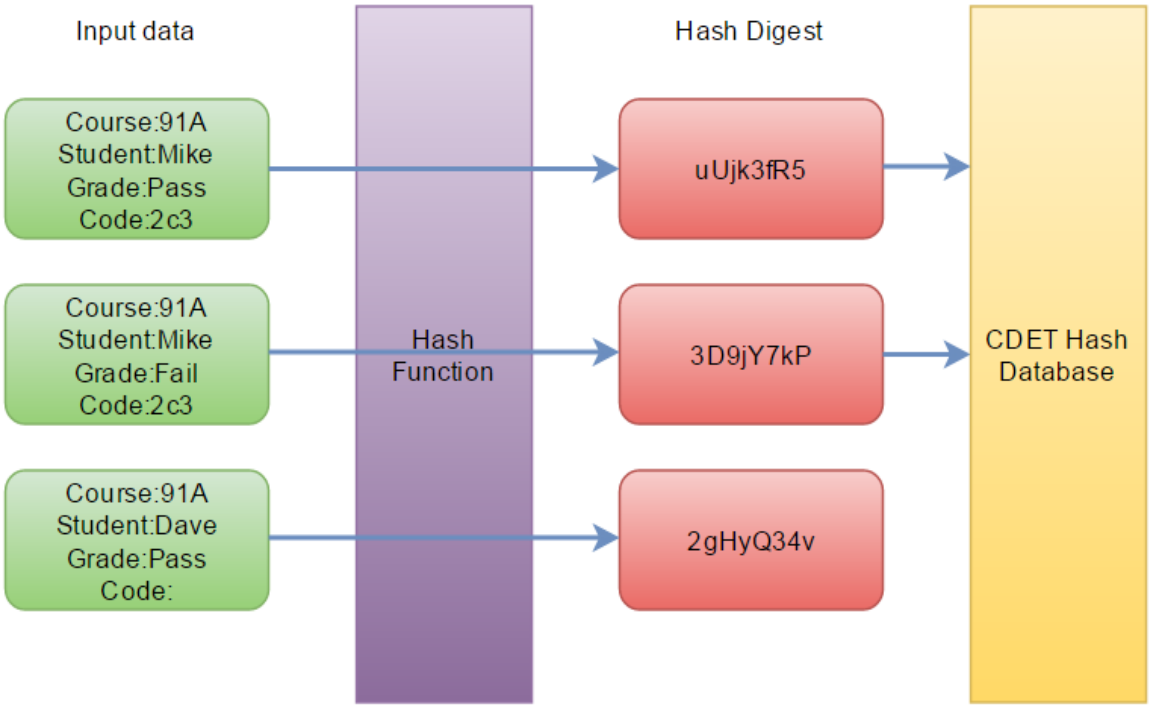


Figure 16. Front-End Work by CDET LRS Storing Pass/Fail Hash.

If the CDET LRS receives hash digest 2gHyQ34v, it will not match any value stored in their database and thus not initiate any further commands to update anyone's transcript. This value could have been falsely submitted by a student using information from Mike's hash, or attempting to submit course completion for 91A under their own name. Because they lacked the download code, no further action could be implemented by the system and thus retaining its integrity.

2. Picture Message

Another option for students to submit a completion message to CDET could be via picture message. By taking a snapshot or screenshot of the completion certificate on their phone, they could simply send that picture to CDET via existing cellular infrastructure. Encoded messages within the picture could be used to verify a user's identity and the information contained in the picture itself could be a Quick Response

(QR) Code, a hash, or another form of completion message that is then extracted by the LRS. By leveraging a personal device's existing capability, there is a multitude of ways to communicate with the CDET learning management system without using an Internet connection.

3. Storing Locally For Future Transmission

If a user would prefer not to send a course completion message to CDET or their circumstances prevent them from doing so, the application will allow them to save their completion message locally until a future moment in time. If a user is limited by their current data plan and would prefer not to use personal resources to transmit their certificates, they will retain the option to wait until interconnectivity is restored or they can transmit a group of messages at once to limit the overall bandwidth used. Once a course is completed, the user will be presented with distinct options and will be able to decide what option works best for them and their current situation. All completed course information can be accessed locally through the transcript button located on the main screen.

G. TRANSCRIPT

The eLearning application must store information that can be referenced by the user. For instance, if young Marines were asked to complete their annual Information Assurance course and submit proof of completion to their Platoon Sergeant, all they would have to do is pull up their eLearning transcript from the application and present to their Platoon Sergeant for unit tracking purposes. It is important that this feature on the main screen be functional in both online and offline modes.

While connected to the Internet, the transcript portion of the application can communicate with the CDET learning management system to update both systems' records. This would allow a Marine to conduct training from their Personal Computer and allow their mobile phone to have a complete list of courses and grades received. Without this communication, a Marine's personal device would not be able to track courses completed on a different platform or know what grades were received on previously submitted course work. Because most training may still be conducted on a PC, it is

important for the mobile device to receive updates to its locally stored transcript. This feature can also work in reverse. If courses are completed offline and the user selected to store course completion messages locally and not transmit at the time of course completion, the transcript portion of this application will be able to communicate with the LRS once connectivity has been restored and would be able to update a student's complete record all at once. This would ease the burden on personal data use and prevent unnecessary overage charges from being incurred by a studious Marine.

The transcript portion of the application will need to separate courses into two categories: enrolled and completed. This feature would allow the Marine to quickly see what courses are currently loaded onto their personal device, what courses they are actively enrolled in, and what courses they have completed and registered with the LRS. By clicking on "enrolled" a user will be directed to a list of courses he or she is registered in and will also inform them of the percent complete status of the listed courses. This information will need to be stored on the device while a student is in the process of taking a course and pulled from the device's memory.

Due to memory restrictions on personal mobile devices, a user may elect to delete locally stored courses once they have completed it. This option will be available in both online and offline modes from the transcript feature of the application. When a user has selected a course from either the enrolled or completed category, the app will simply prompt the user to select which course they would like to delete from local memory. This feature will be very important if the user is operating under limited storage and would like to retain functionality of their phone or download more courses. Memory management must be easy to accomplish.

Under the completed tab of the "transcripts" option, a student will be able to see a full list of the courses they have taken, the grade they received, and the transmission status of the certificate or completion message. If a record has not been updated by the CDET LRS a student can select to transmit from the fourth option on the main screen. Once this retransmission is successful, their eLearning application's transcript will reflect the change in status. Verification with the LRS will happen in online mode only, but will occur frequently to ensure the student's mobile transcript is up to date and current. A

simple red or green indicator could be used to show if a completion certificate had been successfully received and processed by CDET's main record management system. In offline mode, the user should be made aware that their transcript is not currently communicating with the main record system and thus could be inaccurate.

H. TRANSMIT

The eLearning application can communicate with the LRS in real time while connected to the Internet. However, there will be times when a user wants to be able to control when information is passed and in which manner. Transmission options will be presented to the student at the completion of a course, but they are not required to execute them at that time. This initiates the need to have the fourth option on the main screen be dedicated to the transmission of information to CDET. If a user previously completed a course and decides to send their completion message to CDET all they have to do is log into the application, click the transmit button and proceed to make their selections.

Preparation to transmit can take place in offline mode. A user can predetermine how they want to transmit their completion messages to the main learning management system and at what time they want this to happen.

To transmit a certificate, students select which mode they would like to send the message in. The options presented to them will be emailing a JSON message, texting the hash, or submitting the picture message containing a QR code or hash screen shot. Once their selection is made, they will be directed to a page that resembles their transcript in order for them to select which course they would like to send information on at this time. After the course selection has been made, users will be presented with an overview of their selection stating which course was to be selected and transmitted in which mode. They will also be presented with the option to send this transmission now, store this transmission locally, or send once connectivity is restored. These options are critical to ensuring the Marine uses only the data he or she intends to use and CDET minimizes the opportunity for Marines to incur unintended data messaging overages from their mobile phone or data provider. After all sections are made, the user confirms his or her choice and the transmission is sent or stored.

I. HASH DIGEST CONCEPT PROTOTYPE

The goal for our prototype was to identify the feasibility and functionality associated with using a hash function to communicate a course completion message to the learning management system. There are three things that CDET will need to have in place for the hash digest to work. First, CDET needs to have a database capable of storing information from students as they download courses to the mobile app. This database will be used to compare incoming hash digests to hash digests stored when courses were initially downloaded. Second, at the completion of a course, software needs to be in place that will use a hash function to process a specified data set. Lastly, CDET will need to be able to receive an SMS message from students and the hash digest received via text must be compared to the database in order to identify what information should be updated in the LRS.

1. First Attempt

Our first attempt at proving the hash digest concept involved using code generated from www.easygenerator.com. This website was used to create an online sample course for both the server-based and server-less portions of this project. It provided a quick and easy online course that would report completion and user data to the local LRS. In theory, we wanted to download the course constructed for the server-based solution from the web server on to a personal mobile device, execute the course, and have the course produce a hash once it was completed on the phone's web browser. This hash would be stored locally either in the user's notes, picture messages, or simply handwritten onto paper for later submission. This course was a carbon copy of the one designed for the server-based solution with the exception of the hash generation. The server-based solution operated using JSON messages between the course page and the locally hosted LRS.

This attempt proved to be difficult due to the fact that the course was hosted on a website and all html files created upon course download were optimized. Optimized code is code that is automatically generated, in this case via the Easy Generator website, and is written in a manner that tries to reduce the overall size of the file by truncating and oversimplifying various programming methods. Moreover, when code is optimized, it

becomes very difficult to try to discern and make sense of. We attempted to piece together function calls from multiple JavaScript files in order to determine where the contents of the JSON message were being generated and processed. This would have been the location where we could have edited the source code and included a hash function and other local storage commands. Rather than seeing easily identifiable function calls or methods like “def parse_answer” or “def write_JSON,” we instead found “def A,” “def B” and “def C.” It became increasingly obvious that in order for us to locate and identify where specific elements of code were located, we would need to get intimately familiar with thousands of lines of code that were written using complicated abbreviations instead of clear, understandable functions. An automated computer program hosted by a professional site wrote the course’s code upon generation, thus leading to the added complexity. We realized that trying to parse through the numerous course files in order to extract information from the JSON message and correctly implement the hash function would be almost impossible. We wanted to show cross-solution functionality between the server-based and server-less solutions, but it was clear this would not be the preferred course of action to do so.

2. Second Attempt

As we were contemplating our next method to prove the hash digest concept feasible, we decided to show functionality by writing a sample course in Python. We wrote a program that executed similarly to the course generated by www.easygenerator.com for the server-based solution that took a user’s first name, last name, rank, course name and performance score (pass/fail) and created a hash value using a rudimentary hash function.

To show the functionality of this on a personal mobile device we found a web-based application called Trinket.io that allows a user to write and execute Python code while in a web browser such as Firefox, Google Chrome, Internet Explorer, or Safari. Trinket.io even allows users to share code via email, Twitter, Google+, Facebook, and SMS text message via a given link or embedding on their own hosted website. The course

code can be created in Trinket.io using simple Python code and shared with any user to run on his or her personal device's web browser.

There were many limitations with this method. Because this executable Python code was browser-based and no actual file was created, the course code could not be saved locally on a personal mobile device. The only way a user could feasibly run the course on their personal mobile device without cellular or network connectivity is to receive the shared course while connected, and then proceed to complete the course offline while never closing that particular web browser tab. The course could be completed and would generate a local hash, but would be lost should the user close their browser prior to completion while disconnected from the network. He or she would not be able to re-retrieve the course until they reconnected to the network. The moment the web browser is closed this method ceases to be a functional proof of concept.

Trinket.io does not support a multitude of Python libraries. Because the implementation of our server-less solution incorporated the use of robust and well-defined hash functions, we needed to import and use the "hashlib" library from Python. Trinket.io does not support this so we were unable to deliver a legitimate SHA-256 hash function in the Python code to run on Trinket.io. We created this course to showcase a working proof of concept for using a hash function inside the course code. To work around the SHA-256 limitation we manually created a hash function inside our code and were able to successfully produce a hash digest upon course completion. We recognize our self-created hash function was extremely weak and would undoubtedly produce collisions after a very finite number of inputs. The solution proved that a hash digest could be generated from specified data upon course completion, but it is simply not a feasible long-term server-less solution for eLearning.

3. Third Attempt

Knowing we found a way to produce a hash digest from a generated course encouraged us to find a more permanent solution for this study. A solution that could support a legitimate hash function that produces a collision resistant table of hashes would solidify the proof of concept for the server-less option. We decided to see if

someone had developed an established iPhone/Android application available in an application store that supports Python scripts. Our research identified two applications that support our needs. The first is called QPython and is available for free through the Android Apps on Google Play and the second is Pythonista through the Apple Store for a one-time $9.99 fee.

Once we downloaded these applications on our phones we were able to run the same Python program from before with a legitimate SHA-256 hash function. The hash function took input data such as course name, student ID, and score and produced a unique SHA-256 hash that could be submitted and compared to a database containing pre-stored hash values. It is important to note that the Python course we created for our proof of concept requests the necessary user information for hash generation after the course is downloaded instead of before. One reason for this was because we wanted to show multiple different user test cases through the application of one course. The biggest reason though was because we did not possess the necessary learning environment setup to properly demonstrate a user registering for a course and then downloading that particular course with their information imbedded for hash generation. Further research and development can build upon our current setup to help better illustrate the pre-download course registration process that is integral to the creation of the completion hash.

We purchased Pythonista on an iPhone 5S for this proof of concept and found that due to security and proprietary reasons Apple made it very difficult to import pre-written Python files into Pythonista. The work around for this is to purchase another application called Workflow, also available in the App Store for an additional $2.99. Workflow allowed us to essentially bridge the gap between Apple's strict security policies by importing Python code into Pythonista. We simply created a Workflow that went onto an external Dropbox account, copied the contents from the eLearning Python course file onto the operating system's clipboard, and then promptly opened the Import.py file in Pythonista allowing us to paste the code. This copied our program in its initial form, and successfully imported to a new file without the contents being altered.

At this point we came across a minor issue with our course code, specifically a "NameError" issue. Part of our code asks the user to input his or her first name, last name, and rank prior to beginning the e-course. This was producing an error message that we were able to identify and troubleshoot. Pythonista only supports Python 2.7.10 and our original course program was written in Python 3.4.3. After making minor adjustments to our syntax to accommodate the version change, we were able to run our program perfectly as expected and generate a unique pass and fail hash digest for each user.

The major drawback for using the Pythonista application approach was developing the workaround to import a file to prove functionality. However, once this was conducted we were able to run the course disconnected from the Internet and receive a legitimate hash digest.

4. Database

The final step in proving the functionality of this concept is to generate a table of hashes with corresponding user names, course names, and scores and to simulate incoming SMS messages to an LRS. The database was simulated in excel as an example of what the CDET LRS would have to generate and store when a student downloads a course to his or her personal mobile device via the proposed mobile application.

Once we established a working database of 20 users with course names, scores, and hash digests, as illustrated in Figure 17, we were able to run our program via Pythonista using a random registered user name. After completing the course via our mobile device and transcribing the given hash digest, we used SMS to text the hash digest to the simulated LRS. In this case one of us was the receiving node and would physically compare the received text message to our database of hashes to identify which simulated user had completed a specific course and what his or her completion status was.

Last Name	First Name	Course ID	Score	SHA-256 Hash
Benson	Joshua	IA2016	Pass	d107c4c7b51d14fa83484e1221cfa9c28787faf94a3e6f925571431322aaea02
			Fail	3849f5986751d523acdb34feafdc36a5ef480bd97ac4423808f2a22866a35bf9
Greunke	Brian	IA2016	Pass	6460bf7fefc005a429f429569eb580696ff9e25b358332b8a00e8c25c2b86050
			Fail	691a7205ae057320a5f307182fe3368191c99137e3c81e15a6415f4e983431ce
McCarthy	Brian	IA2016	Pass	b0d0e7a7f80c4ce8923503eb17e035774bf0759f54626c0cb76fa34cbf117238
			Fail	b3bf56154942759c2d6fe6162135ac0ba8d008a8107f58481cedb1e3d6015b17
Crawford	Kirk	IA2016	Pass	a9894f3b596285f5f57cfe66c1b713421de8c437a8fe353738be6926d8154863
			Fail	3cd997aac4deb381e03ce240744b1c9a329be7daa8ceffe7671e9718cca903ae
Benson	Joshua	SAPR16	Pass	2f61fc8400ea8274015a05645f7ca5e995897c7df0d987cdbd943f223e53f7d7
			Fail	7908403876f16001320e23fed0ed9a20ea02cc4e7627a012ee9cb3d6f1d98539
Greunke	Brian	SAPR16	Pass	346a6da5b5a377cf2c18e4e923a0f76214f35e005cbb2621c140670d349f1ad9
			Fail	d9aedc9437553e029189e32b6142ea09b14dab8052df6fa06d6aa51518f248ed
McCarthy	Brian	SAPR16	Pass	26e94d496a1a1d76632e817ed598629fae1cf0d4f0d73007551afbd1fca76431
			Fail	d1ceca6095462b93de81755935f8278adcb50ad15e20fbe5916b5315fa126035
Benson	Chrissa	Money101	Pass	9be446c8df9a85b8c36c79ae036b2bc2cba63f98eb13e6feb849f9412d56c123
			Fail	dea8af7ff2d6fc0b8e946b29f0da238f6168e4d19b5cc066f813223e7966aef1
Spieth	Jordan	Golf101	Pass	e2048a5bff2749064922ba8a66efc904195e231153eb34814516391aa87bfe91
			Fail	55ca7bde40ee572320abb06b503f44dd51472b4f5f99a01e1df70e8bc9c9f72c
Spieth	Jordan	IA2016	Pass	66d8628527894d13f8df9e1a797b51a690f725f4ac8384a82d73dc7a17cbdd49
			Fail	78de22195402751b95a72953638d608c80ef88ff455928d71a600cb5073d3ca1
McCarthy	Meredith	IA2016	Pass	7ea5602c3b1767ecce580ef21696ab133ee87c1eb89ad6bcb1bd91693262ea1e
			Fail	b5bba85e2c5af505ed4161dabff17de5f4e27a9ed8ffb1cd94f07e73a1d1da83
Collier	Anthony	Shoot101	Pass	9acb41add291dae8dbed44b12679adb3325beef0719761223f746313b0d01a7c
			Fail	ff56465fb3fb8122ecb337924bd8373d50ebd815740c7cb7187cf5f8931f2ad9
Collier	Anthony	IA2016	Pass	53a505b51b0a116b040fa303b2d29faf63930725f3a2e242f4bf8eaa8d276615
			Fail	4348a659367e99a8246943739f847d92f896bb1e8c35df1ec92836b768d5484f
Collier	Anthony	Jumper23	Pass	1bca74233c8539840b04e07845f39102ed5a1c37941b0c3ffcd2d37f7c1f9cfa
			Fail	b28e989edc6ed351e63931f8b75c59e2c406d81cf9dd404eadcf789d06bd095f
Anderson	Erik	IA2016	Pass	5fa69583c6c9f8c8e723af16d688d1a119031bfc7dbefbd105b0482d651acd10
			Fail	44131a919449d2e56f0053d00b2cf580da3f0ce9e841fc5dee96578bcb3c2e23
Anderson	Erik	SAPR16	Pass	9de666ea20abede2c373f9a5bb6969a94d06ac7d62d778cc5f00f0ac02ae692c
			Fail	a70f079a305d7d3d1b4f9bf71b64c2f1eb4c9c79ac3c76e05391e5f8ee8736ee
Anderson	Erik	Golf101	Pass	df0541ff61fe54f9aa87f869812a32d0555cedba2ca461a8bdfa64a770163a73
			Fail	78a5a4278dfc8a0e1440c5b8c316585285ad93f4d39d79e897c3f3bbf567cce0
McCarthy	Brain	Golf101	Pass	fcb8fc0fd1f54643ae4eda8583cbe07787687fd576e774e4dfbf9a90a606bb84
			Fail	fad8279215a89193bba21854727aa7a491dd6f1b51cf43a894f77e5c4a169188

Figure 17. Snap Shot of Mock CDET Database

When a student downloaded a course, his or her first name, last name, course, and potential results were added to the database. Specific information was then hashed using a SHA-256 hash generator. We used a simple online hash calculator located at http://www.xorbin.com/tools/sha256-hash-calculator. The format for submitting the pre-hashed data followed this very specific format: Firstname, Lastname, COURSENUMBER, Result. The result was either “Pass” or “Fail” for demonstration purposes. The generated hash digest was then added to the student’s information in the database. For example “Brian Greunke, SAPR16, Pass” was passed through the SHA-256 generator to produce the following hash digest:

346a6da5b5a377cf2c18e4e923a0f76214f35e005cbb2621c140670d349f1ad9

5. Test Cases

Once we established a large enough database with hash values associated with each name, course and score, we began testing our concept using the Pythonista program.

Our first test case was conducted via the mobile device in airplane mode to demonstrate the program's ability to function while disconnected from the Internet. As seen in Figure 18, a student participated in a course from his or her mobile device and received a hash for course completion.

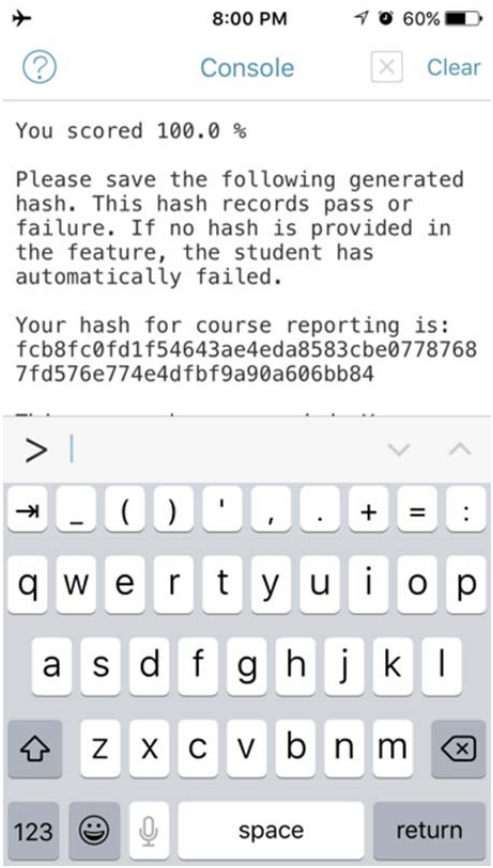


Figure 18. Screenshot of Course Completion Hash

While in separate locations to ensure complete isolation between systems, a text message containing the hash digest was sent to the other person with the database open. Without any communication other than the hash, the person observing the database is

able to determine that Brian McCarthy completed the Golf101 course with a passing grade. As seen at the bottom of Figure 17, hash value fcb8fc0fd1f54643ae4eda8583cbe07787687fd576e774e4dfbf9a90a606bb84 is a "pass" registered to Brian McCarthy for downloading the Golf101 course.

Our second test case demonstrates the registration process and completion message for a student on his or her mobile device in airplane mode. Instead of receiving the user's information before download and imbedding this information in the course for hash generation purposes as presented in the proposed solution, the setup of our proof of concept prompts the user for registration information after the course is downloaded. Figure 19 demonstrates the user interface from a mobile device upon course registration, and the end result of course completion. Once the student receives the hash digest after the course is finished, he or she would then send this unique hash to CDET to report their course completion status.

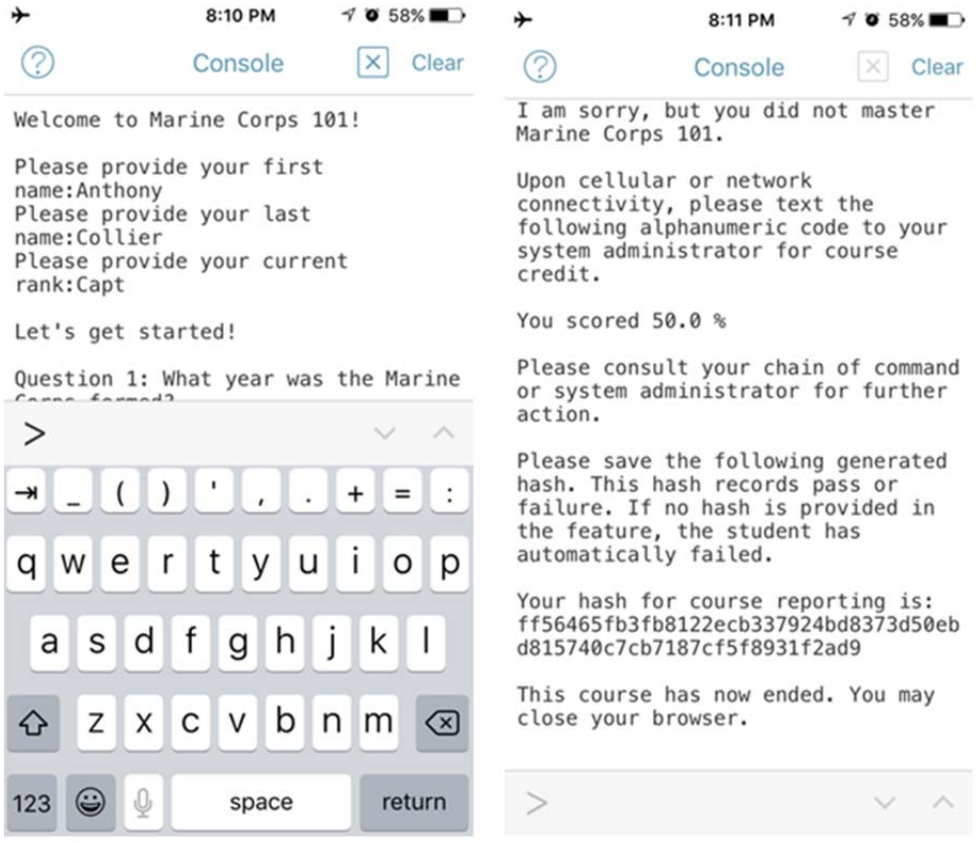


Figure 19. Screenshots of Mobile Course Registration and Hash Generation

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION AND FUTURE WORK

A. CONCLUSION

The intent for this thesis was to bring to light two possible solutions that could help integrate the BYOD construct into the Navy and Marine Corps' respective eLearning environments. Both solutions, while currently limited due to their infancy, provide a good baseline to continue to build upon. The aforementioned proofs of concept illustrate how a service member, with little to no technological experience, can access or download a particular course, take the course, and receive credit for the course all while using their own personal device. The challenge moving forward will be twofold. The first is scalability. While our solutions proved to be successful in a vacuum, they were conducted at a very small scale with a very small footprint. As the number of users swells from one to thirty, and the number of courses increases from one to forty, so does the task of implementing these growths in our given solutions. For the server-based solution, this influx can substantially affect the performance of the local wireless network and the ability to deliver larger content to more users. For the server-less solution, considerations need to be made regarding file size and how much a service member can be expected to save locally on their device. The basic Python script we implemented to demonstrate a simple evaluation is one case, but what about the countless courses that utilize extensive graphics and require increased user interaction? This, of course, will require the implementation of the mobile application discussed in chapter four, but these are all considerations and known unknowns that will have to be addressed in the near future.

The second challenge, which is easily the most difficult one, is to get DON and DOD leadership to buy into the BYOD movement and accept the necessary risks associated with its implementation. The extent of a BYOD-friendly environment in the military would only touch the eLearning realm, and would be isolated and segregated to prevent any form of bleed over into official government business. This isolation using dedicated assets is the solution to a problem that, deservedly so, has Pentagon brass extremely skeptical and hesitant. If the DON wants BYOD to be the future of eLearning and serve as the force multiplier they are looking for, then the legwork, both from an

acquisitions and policy perspective, needs to be done diligently. Risk aversion, while understandable, is not the answer and needs to be quelled.

B. FUTURE WORK

The following subsections are just a handful of topics we believe need to be explored in greater detail in order to further the research and development of the proposed eLearning BYOD solutions. Some topics are much broader in scope than others, but they need to be addressed nonetheless. This list is not all-inclusive, and we realize there are areas of research we are not attuned to that would assist in these recommended solutions truly coming to fruition.

1. Extensive Load Balance Testing for Wireless Network

The server-based solution explored in this thesis used a very powerful wireless router, but only implemented one user as a test case and a very basic three-question course. To display true network functionality, the network needs to be expanded and simulated with multiple different devices accessing a wide range of content at the same time. Increasing the number of users and allowing the server to host more content will better illustrate the capabilities of the network and identify any limitations that must be addressed prior to pursuing a formal solution. The next step is replication of the server-based proof of concept, but at a much larger scale in order to depict how a real-life BYOD eLearning environment would behave.

2. Policy Surrounding Shipboard Wireless Networks

Because the server-based solution operates strictly using a wireless network connection, the underlying code of federal regulations (CFRs) and policies need to be explored, or possibly developed, concerning the establishment of a wireless network on board ship while underway. What wireless security measures must be taken and implemented when using the server-based wireless network? Are there certain operational conditions where the usage of a wireless network is prohibited? What RF considerations must be taken into account when employing a shipborne wireless network? Will a wireless network have any negative implications on other ship-based

communication, navigation, or targeting assets? These are all important topics for discussion and issues that need to be addressed should a server-based solution be deployed at sea.

3. Mobile Course Prototype

For BYOD in eLearning to become a reality for CDET and the DON, current course offerings must be made available for PEDs. CDET confirmed that an attempt to create a mobile device capable course was initiated, but that it is not currently an active course on MarineNet. As follow-on research to build on the concepts identified in this project, it is important to examine the feasibility and requirements necessary to convert active MarineNet courses into mobile capable eLearning modules. The main goals of the BYOD eLearning concept is to be able to access required content and conduct online training from anywhere a mobile device can function. Lessons adapted for mobile devices will allow the BYOD eLearning concept to expand and encourage further development in terms of an application's user interface, reporting options, and back end support modifications.

4. User Interface Prototype

In order for eLearning to take place on a personal mobile device, it is crucial to develop a working prototype of the application laid out in the research of this project. Future work to develop an application capable of functioning on a variety of personal devices and that accounts for the heterogeneity of mobile computing would allow BYOD eLearning to expand from an idea to a reality. The application must address local memory storage options, completion message transmission options, and authentication requirements that satisfy the DOD policies and regulations.

5. Blackboard Mobile

CDET currently uses MarineNet as their primary eLearning platform, which is a Blackboard-based product. Blackboard offers mobile versatility that could be used in place of creating a brand new mobile application from scratch. Future research into the feasibility of offering Blackboard mobile to MarineNet users could alleviate the need to

produce a new user interface. Questions that need to be addressed are: Can Blackboard mobile offer a “no Internet connectivity” reporting option? Can Blackboard mobile host current MarineNet classes? Which option for a mobile platform is financially responsible for the DOD to consider for long-term use?

6. Virtual Machine

Some research has been conducted to determine the versatility and functionality virtual machines could provide to the eLearning environment. More research can be conducted to blend the mobile device application with server-based virtual machines. This option could help alleviate the need to create mobile device-specific course content and instead turn the phone into a simple mobile screen for a personal computer operating elsewhere.

7. QR Codes

The QR code is a new and reliable upgrade to a one-dimensional barcode. With the advances in technology and the ever-increasing demand for a higher capacity coding scheme, the two-dimensional barcode was born. The two-dimensional barcode has been named the QR code and is considered two-dimensional because of its ability to store information both vertically and horizontally. The QR code drastically increases the amount of data that can be stored compared to one-dimensional barcodes. QR codes offer a plethora of storage potential for transmitting data.

For use in the eLearning application we are presenting with this study, it would be feasible for a completed course to produce a QR code that stores similar content to a JSON message without the need to transmit in paragraph form. The course name, student, scores, and time stamp could easily be programmed into the QR code to be transmitted to the CDET learning management system via email or picture message directly from a smart device. Follow-on research should consider the possibility QR codes present for tracking eLearning when disconnected from the Internet.

APPENDIX. PYTHON COURSE SOURCE CODE

```
1.  from __future__ import division
2.  import hashlib
3.
4.  # Marine Corps 101 Sample Course
5.
6.  #Course banner
7.  print ”\nWelcome to Marine Corps 101!\n”
8.
9.  #Stores the user’s first name, last name, and rank for record keeping
10. fname = raw_input(“Please provide your first name:”)
11. lname = raw_input(“Please provide your last name:”)
12. rank = raw_input(“Please provide your current rank:”)
13. fullname = (rank + ” ” + fname + ” ” + lname)
14. hashname = (fname + ” ” + lname + ,” Golf101, ”)
15.
16. #Keep track of correct and incorrect responses
17. incorrect = 0
18. correct = 0
19. total = 0
20.
21. #Begin the course
22. print ”\nLet’s get started!\n”
23.
24. #Question 1
25. print ”Question 1: What year was the Marine Corps formed?\n”
26. answerone = raw_input(“Please provide your answer in four digit form:”)
27. if answerone != ”1775”:
28. print ”\nThat is not the correct answer.\n”
29. incorrect +=1
30. total +=1
31. else:
32. print ”\nCorrect!\n”
33. correct +=1
34. total +=1
35.
36. #Question 2
37. print ”Question 2: Where was the Marine Corps formed?\n”
38. answertwo = raw_input(“Please provide your two word answer:”)
39. if answertwo == ”Tun Tavern” or answertwo == ”tun tavern” or answertwo == ”Tun tave
    rn” or answertwo == ”tun Tavern”:
40. print ”\nCorrect!\n”
41. correct +=1
42. total +=1
43. else:
44. print ”\nThat is not the correct answer.\n”
45. incorrect +=1
46. total +=1
47.
48. #Calculates overall score
49. score = ((correct/total)*100)
50.
51. #End of Course
52. if correct > 1:
53. status = ”Pass”
```

```
print ”Congratulations, ,” rank + ” ” + lname, ”! You have mastered Marine Corps 101.\n”
#print ”Your name and results have been stored locally on your phone under the text file ’CourseResults.txt’.\n”
print ”Upon cellular or network connectivity, please text the following alphanumeric code to your system administrator for course credit.\n”
print ”You scored,” score, ”%\n”
else:
status = ”Fail”
print ”I am sorry, but you did not master Marine Corps 101.\n”
#print ”Your name and results have been stored locally on your phone under the text file ’CourseResults.txt’.\n”
print ”Upon cellular or network connectivity, please text the following alphanumeric code to your system administrator for course credit.\n”
print ”You scored,” score, ”%\n”
print ”Please consult your chain of command or system administrator for further action.\n”

#SHA-256 Hash of Results
arg = hashname + status
arghash = arg.encode(‘utf-8’)

print ”Please save the following generated hash. This hash records pass or failure. If no hash is provided in the feature, the student has automatically failed.\n”

print ”Your hash for course reporting is:\n,” (hashlib.sha256(arghash).hexdigest())

#Makeshift hash function. MD5 and SHA1 libraries are not supported by trinket, but this is where they would be implemented.
##arg = hashname[::-1]
##if status == ”Mastered”:
## code = arg[-1] + ”p” + arg[0] + ”d” + arg[-2] + ”7” + arg[4] + ”r” + arg [-4]
## print (“Your alphanumeric code is: ,” code)
##else:
## code = arg[-3] + ”h” + arg[2] + ”j” + arg[-1] + ”3” + arg[0] + ”r” + arg [-2]
## print (“Your alphanumeric code is: ,” code)

#End of course
print ”\nThis course has now ended. You may close your browser.”

#Writes the results and the users information to a text file locally on their device
#f = open(“CourseResults.txt,” ”w”)
#f.write(“\nCourse Results Identifier: ” + code)
#f.write(“\nServicemember: ” + rank + ” ” + fname + ” ” + lname + ”\nScore: ”)
#f.write(str(score))
#f.write(“\nPerformance: ” + status)
#f.close()
```

LIST OF REFERENCES

Defense Information Systems Agency. (2015, June 10). Strategic Plan 2015–2020. Retrieved from http://www.disa.mil/~/media/Files/DISA/About/Strategic-Plan.pdf.

Defense Information Systems Agency. (n.d.). DOD Mobility Unclassified Capability. [Online]. Retrieved from http://www.disa.mil/Enterprise-Services/Mobility/DMUC

Defense Information Systems Agency. (n.d.). Identity and Access Management. [Online]. Retrieved from http://www.disa.mil/Initiatives/Identity-Access-Mgmt

Department of Defense. (2012, February 24). DOD Information Security Program: Overview, Classification, Declassification. Retrieved from http://www.dtic.mil/whs/directives/corres/pdf/520001_vol1.pdf.

Department of Defense. (2012, June 8). *Department of Defense mobile device strategy* [Memorandum]. Retrieved from http://archive.defense.gov/news/dodmobilitystrategy.pdf

Department of Defense. (2013, February 15). *Department of Defense commercial mobile device implementation plan* [Memorandum]. Retrieved from http://archive.defense.gov/news/DoDCMDImplementationPlan.pdf

Department of Defense. (2014). 2014 Demographics Profile of the Military Community. Retrieved from http://download.militaryonesource.mil/12038/MOS/Reports/2014-Demographics-Report.pdf.

Department of Defense. (2014, March 12). *Risk management framework for DOD information technology*. (DOD Instruction 8510.01). Retrieved from http://www.dtic.mil/whs/directives/corres/pdf/851001_2014.pdf

Department of Defense. (2014, March 14). *Department of Defense cybersecurity instruction*. (DOD Instruction 1000.13). Retrieved from http://www.cac.mil/docs/DODI-1000.13.pdf

Department of the Navy. (2007, August 20). *DON security guidance for personal electronic devices (PED)* [Memorandum]. Retrieved from: http://www.doncio.navy.mil/ContentView.aspx?id=347.

Department of the Navy. (2008, April). Enterprise Mobility 2008. Retrieved from http://www.doncio.navy.mil/uploads/1024ELU32586.pdf.

Department of the Navy. (2012, February 27). *Memorandum for the Department of Defense Chief Information Officer* [Memorandum]. Retrieved from http://www.doncio.navy.mil/uploads/0612XPQ94367.pdf.

Gondree, M. (2014). “Integrity.” CS3600 [PowerPoint]. Monterey, CA: Naval Postgraduate School, 32–55.

Grim, N. (2013, July 26). Marine Corps Mobile Device Strategy Looks to Cut Costs. [Online]. Retrieved from https://defensesystems.com/articles/2013/07/26/marine-corps-mobile-device-strategy.aspx

Hinum, S. (2012, May 26). Dell Latitude E6500. [Online]. Retrieved from http://www.notebookcheck.net/Dell-Latitude-E6500.12268.0.html

Linksys. (2016). Linksys WRT1900AC AC1900 Dual-Band Smart Wi-Fi Wireless Router. [Online]. Retrieved from http://www.linksys.com/us/p/P-WRT1900AC/

Microsoft. (n.d.). Learn About Windows Hello and Set It Up. [Online]. Retrieved from http://windows.microsoft.com/en-us/windows-10/learn-about-windows-hello-and-set-it-up

Paradigm. (2008, January 2). Metadata for Authenticity: Hash Functions and Digital Signatures. [Online]. Retrieved from http://www.paradigm.ac.uk/workbook/metadata/authenticity-fixity.html

Scarfone, K., Jansen, W., & Tracy, M. (2008, July). *Guide to General Server Security*. (Special Publication 800–123). Retrieved from http://csrc.nist.gov/publications/nistpubs/800-123/SP800-123.pdf.

Smith, A. (2015, April 1). Pew Research Center: Internet, Science and Tech. [Online]. Retrieved from http://www.pewinternet.org/2015/04/01/chapter-one-a-portrait-of-smartphone-ownership/.

Smith, L. E. (2015). MarineNet statistics MarineNet stats through December 2014. In *Technical Director Brief on MarineNet Statistics.* Quantico, VA: Education Command, College of Distance Education & Training.

United States Marine Corps. (n.d.). United States Marine Corps College of Distance Education and Training. [Online]. Retrieved from https://www.mcu.usmc.mil/cdet/SitePages/home.aspx

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California