NPS-IT-22-246

ACQUISITION RESEARCH PROGRAM
SPONSORED REPORT SERIES

Exploration of Software License Management Methods in Government Enterprises

June 2022

Capt James A. Hughes, USMC

Thesis Advisors: Anthony Canan, Assistant Professor
Glenn R. Cook, Senior Lecturer

Department of Defense Management

Naval Postgraduate School

Prepared for the Naval Postgraduate School, Monterey, CA 93943

ACQUISITION RESEARCH PROGRAM
DEPARTMENT OF DEFENSE MANAGEMENT
NAVAL POSTGRADUATE SCHOOL

The research presented in this report was supported by the Acquisition Research Program of the Department of Defense Management at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact the Acquisition Research Program (ARP) via email, arp@nps.edu or at 831-656-3793.

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School

ABSTRACT

As organizations expand their information technology (IT) infrastructure, the task of properly managing third-party software licenses becomes increasingly challenging. To better manage this problem, it is critical to understand the underlying need for licenses, identify effective IT governance policies, and determine capable software discovery tools. Through a series of interviews with experts working in the fields of IT and asset management, the best practices and policies currently in effect were explored to determine how a large enterprise should approach software license management (SLM). Findings indicated that while technological solutions can assist with providing awareness of what exists on an organization's network architecture, they are overall insufficient in improving SLM across multiple networks. Enterprises must consider those factors that increase the risk for individuals to use unauthorized software or circumvent license agreements. Through managing these risk factors and applying more radical methods such as introducing software auditing teams or delegating SLM responsibility altogether, enterprises may find themselves more readily capable of avoiding the legal and financial costs of breaching software license agreements.

The research presented in this report was supported by the Acquisition Research Program of the Department of Defense Management at the Naval Postgraduate School.

To request defense acquisition research, to become a research sponsor, or to print additional copies of reports, please contact the Acquisition Research Program (ARP) via email, arp@nps.edu or at 831-656-3793.

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Mustafa Canan, for taking the time to mentor and advise me throughout the entire research process. Your guidance helped me look at problems through different perspectives that I otherwise would never have discovered on my own.

Most importantly, I never could have achieved any success without the love and support of my wife, Christa Hughes. Despite how busy our lives became, you were always there to help me stay strong and keep moving forward. Thank you for tolerating my very busy schedule and for everything that you have done to help make our time in Monterey memorable.

THIS PAGE INTENTIONALLY LEFT BLANK

NPS-IT-22-246

ACQUISITION RESEARCH PROGRAM
SPONSORED REPORT SERIES

Exploration of Software License Management Methods in Government Enterprises

June 2022

Capt James A. Hughes, USMC

Thesis Advisors: Anthony Canan, Assistant Professor
Glenn R. Cook, Senior Lecturer

Department of Defense Management

Naval Postgraduate School

Prepared for the Naval Postgraduate School, Monterey, CA 93943

ACQUISITION RESEARCH PROGRAM
DEPARTMENT OF DEFENSE MANAGEMENT
NAVAL POSTGRADUATE SCHOOL

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

BMC	baseboard management controller
BSA	business software alliance
BYOD	bring your own device
CES	consolidated engineering services
CMDB	configuration management database
COTS	commercial off-the-shelf
DOD	Department of Defense
DRM	digital rights management
D-UNS	deliberate universal needs statement
EEVE	Enterprise Engineering and Verification Environment
eMASS	Enterprise Mission Assurance Support Service
ETG	enterprise topology graph
EULA	end user license agreement
FOSS	free and open-source software
FSF	Free Software Foundation
FSMAO	Field Supply and Maintenance Administration Office
GOTS	government off-the-shelf
HBSS	Host Based Security System
HOT	human-organization-technology
HP NA	Hewlett-Packard Network Automation
IT	information technology
ITOM	Information Technology Operations Management
ITACS	Information Technology and Communication Services
LGPL	lesser general public license
LOGCOM	Logistics Command
MCCS	Marine Corps Community Services
MCEITS	Marine Corps Enterprise Information Technology Services
MCEN	Marine Corps Enterprise Network
MCIEAST	Marine Corps Installations East
MCRC	Marine Corps Recruiting Command

MCSC	Marine Corps Systems Command
MCU	Marine Corps University
NASA	National Aeronautics and Space Administration
NCDOC	Navy Cyber Defense Operations Command
NPS	Naval Postgraduate School
OSS	open-source software
PDF	portable document format
PEO	Program Executive Office
PII	personally identifiable information
ROI	return on investment
SCCM	System Center Configuration Manager
SETA	security education training and awareness
SLM	software license management
SONIC	Secure Operational Infrastructure and Communications
TECOM	Training and Education Command
TSO	Technology Services Organization
U.S.	United States
USC	unauthorized software copying
USMC	United States Marine Corps
VM	virtual machine
VPN	virtual private network
ZTA	zero trust architecture

I. INTRODUCTION

A. EXAMINING THE PROBLEM

The growth of the United States Marine Corps (USMC) enterprise and its increased reliance on digital tools has led to increasing demand for software variety to fulfill its mission. Across the numerous networked domains, such as the Marine Corps Enterprise Network (MCEN), multiple asset discovery tools have been deployed to discover and track active software licenses and the utilization of unauthorized software. The discovery tools, which include but are not limited to BMC Discovery, ForeScout, and Tanium, are partially effective in single domains but struggle to consolidate the data across the entire Marine Corps enterprise as these tools do not function across all networks. This issue is further exacerbated by the tendency of users to seek out software solutions that meet their task requirements but have yet to be vetted or authorized for use. Without the capability to fully track and manage the utilization of unlicensed and unauthorized software, the USMC will continue to experience the underlying security and legal ramifications related to its utilization.

B. PURPOSE STATEMENT

This study addresses the capability gap stated by Marine Corps Systems Command (MCSC) in asset discovery management across the USMC enterprise and some of the fundamental causes surrounding unauthorized software utilization. A qualitative approach is used to collect information regarding best practices and procedures for software and license management across various organizations that face similar challenges across enterprises that are similar to that of the USMC. The initial phase of the study involves identifying the organizations and conducting selective interviews with personnel familiar with this problem set in the context of their organization. The subsequent phase is the analysis and exploration of each interview, comparing and contrasting the methods in which each organization handles software discovery, license management, and techniques to mitigate the use of unauthorized software. Within this phase, the intention is to elicit the best management practices and

procedures that the USMC enterprise can feasibly adopt. The proposed framework must be feasible and cost-effective for military enterprises to implement and enable them to manage software licensing across multiple networks.

C. RESEARCH QUESTIONS

1. What underlying factors exist that promote the mismanagement of third-party software and utilization of unauthorized software on enterprise networks?
2. What best practices can be applied to reconcile software license management across an enterprise consisting of multiple networks and domains?
3. What additional standards of enterprise governance should the USMC adopt to best enforce software license management and unauthorized software utilization?

D. RESEARCH METHODOLOGY

This research was conducted utilizing a qualitative study method. A series of interviews were conducted from government enterprises to provide a focused perspective from the viewpoints of experts operating under similar working environments. The initial phase of this study was screening the potential organizations to ensure they meet the appropriate criteria for an enterprise that most closely resembles that of the USMC. These criteria include the number of employees, the number of networks and domains it controls, geographical dispersion, and country of origin. Organizations outside of the United States were not included in this study for security and the legal requirements that the USMC is required to maintain under U.S. law regarding information systems. To ensure that each organization is sufficiently researched, a maximum of eight organizations were to be included in this study. The secondary phase of this study identified and selected individuals within each organization for interviews. At least one interview was conducted for each organization, and interviews were expected to last no longer than one hour. In addition to any documents and correspondence that the

interviewees were willing to share, open-source information regarding each organization's IT practices were explored to supplement interview data.

The final phase consisted of analyzing the data gathered from each interview and the data from open sources. The focus was on the best methods and practices that each organization utilizes to mitigate the use of unauthorized software and its ability to manage software licensing across multiple networks. These tools and methods were compared to those utilized by the USMC to determine which are most feasible to apply and adopt. Due to government-owned networks' legal and security limitations, any software-based recommendations must eventually receive assessment by the appropriate information assurance agencies prior to implementation. This process ensures that the final analysis consists of an acceptable solution for a federal government institution to implement.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND AND LITERATURE REVIEW

A. EXISTING GAPS WITHIN THE USMC

1. MCSC Discovery Limitations

The basis for this study stems from current capability gaps within the USMC, specifically how it can manage asset discovery and software license management (SLM) at the enterprise level. For purposes of this work, the term enterprise refers to any complex system that constitutes individuals or groups who coordinate actions to pursue a common goal (March & Simon, 1958). Figure 1 displays this gap in coverage over the various garrison and tactical edge networks that the USMC operates. Though software management tools such as Tanium, ForeScout, and BMC Discovery can capture a limited amount of data on client and server assets, they lack any sensible means of aggregating this data into a single repository.

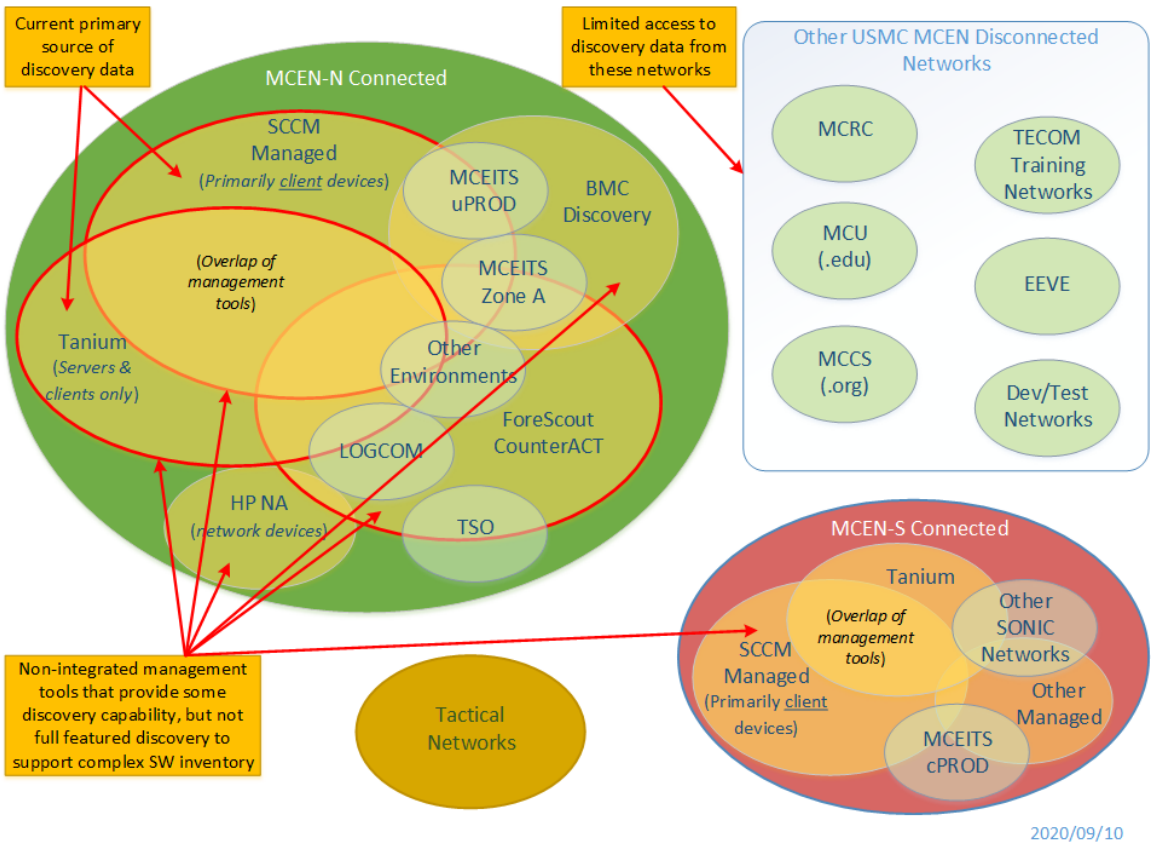

Figure 1. MCSC Enterprise Discovery Scope Challenges.
Source: Toohey et al. (2020).

There is value in determining whether or not a technical means of aggregation of licensing and discovery data exists. Such tools would potentially enable a means to automate the collection process of that data, generating a detailed list of what assets exist on all networks, what applications they run, and even the current state of the licenses of those applications. However, it is also important to take a more holistic view of this problem set and explore the organizational factors that force such measures to need to be put in place. A purely software-driven and technical solution to this problem set, while desirable, is not guaranteed. For example, the USMC consists of over 180,000 active-duty service members, operating in regions across the globe (Department of Defense [DOD], 2018). It is entirely possible that at this scale, SLM and application management, in general, may require adherence to policies and practices that are not solely enforced by technical safeguards.

Strategies of enforcement of policies and practices can be broken into two separate categories: non-technical means, which require written or potentially legal compliance to dissuade certain activities, and technical means that provide mechanical stoppages (Von Solms, 2006). A combination of the two is most likely necessary to prevent unauthorized usage of unaccredited software. Previously developed theories such as the human-organization-technology (HOT) framework explore these relationships in more detail regarding the overall health of information systems (Kumar et al., 2020). This framework asserts that the more compatible human, technology, and organization factors are with one another, the greater the overall health of information systems and cyber security posture will be (Kumar et al., 2020). Prior research by Kumar et al. (2020) revealed that the top antecedents to establishing a strengthened cyber security posture were legal consequences, technical measures, the role of senior management, and proactive information security measures. While these factors relate more directly to cyber security than SLM and unauthorized use of software, certain parallels may be drawn to assist in tackling these challenges.

2. Funding and Cost Considerations

To obtain support for addressing the problem set, Program Executive Office (PEO) Digital, a sub-organization within Department of the Navy, took steps to quantify the problem set into financial metrics. This aimed to specifically address the data consolidation problem and the need for acquiring a subsystem that can aggregate discovery data across the MCEN. The MCEN does not encompass all networks utilized by the Marine Corps; however, it is the largest network used by its forces in garrison.

After assessing the functionality requirements of a data consolidation tool and the acquisition strategy necessary to contract it utilizing Other Transactional Authority (OTA) means, several cost determinations were found. An FY21 pilot program would cost roughly $20,000 to $40,000 for a 90-day period, acquisition and implementation would be $800,000, and sustainment for three following years would cost $705,000 (Scuderi, King & Toohey, 2021). With these potential costs in mind, the next step was to develop a proposed return on investment (ROI) that could display the potential amount of

cost savings. This is difficult to quantify, as cost savings through risk reduction and productivity improvement inherently do not carry a specific value. Figure 2 displays PEO Digital's findings for these potential savings. For example, over a five-year period, it is estimated that by not requiring operational support to conduct manual data consolidation, the USMC could save $6,147,979 (Scuderi et al., 2021). Each category listed in the x-axis represents additional areas that, over time, could experience cost avoidance with the addition of an automated tool or process for aggregating software discovery data.

	5Year Cost Savings Projections from automation of processes					
	Software License Rationalization	Operations and Support Cost	Research and Validation Time	Effort to Manage Internal Audit Prep	Effort to Manage External Audit Prep	Risk of Security Breach
Year 1	$5,361,300	$1,158,000	$8,685,000	$66,465	$27,346	$73,181
Year 2	$5,414,913	$1,192,740	$8,902,125	$68,127	$28,029	$73,181
Year 3	$5,471,073	$1,228,522	$9,124,678	$69,830	$28,730	$73,181
Year 4	$5,529,824	$1,265,378	$9,352,795	$71,576	$29,448	$73,181
Year 5	$5,591,215	$1,303,339	$9,586,615	$73,365	$30,184	$73,181
Total	$27,368,325	$6,147,979	$45,651,213	$349,362	$143,737	$365,905

Figure 2. Data Consolidation Cost Avoidance Sheet. Source: Scuderi et al., (2021).

As such costs are merely speculative, there are no true means to determine whether or not cost prevention is occurring. The size of these figures, however, severely outweighs the cost of the actual system, and even in the first year alone, the overall cost of the system ($840,000) is merely 5.4% of the potential savings ($15,371,292) (Scuderi et al., 2021). Nevertheless, even when speculating that none of these cost savings figures are possible, there is still a benefit in automating the discovery process by removing paying for manual labor. It was estimated that the total cost of manual labor, which includes script development, database administration, and data analysis through contracting and government employees, would reach $939,000 annually (Scuderi et al., 2021). On its own, this already outweighs the costs of purchasing the data consolidation automation tool, which was estimated to be $805,000 annually over 5 years (Scuderi et

al., 2021). The significance of these figures demonstrates the importance of taking steps to automate and improve SLM and software discovery data consolidation processes.

B. UNAUTHORIZED SOFTWARE UTILIZATION

It should come as no surprise that the problem set involving unauthorized software on government or business-owned devices is not uncommon. Choices that lead to this introduce security risks to organizational devices and any connected network. Examining the underlying reasons why users make these choices in the first place can assist with policy development at higher organizational levels. A significant amount of literature regarding these phenomena is worth exploring to understand them better.

1. Ethics and User Circumvention

Software is accepted as a copyrightable work and is thus entitled to particular protections as an intellectual property (Hsieh & Yeh, 2012). However, there exist many applications in which a user can duplicate digital content without the developer, distributor, or organization's consent (Hsieh & Yeh, 2012). This phenomenon referred to as unauthorized software copying (USC) remains an issue that directly influences the need for software license management and software discovery automation tools. Understanding the influential variables that exacerbate a user's tendency to commit USC could, in turn, influence effective policy-making decisions that dissuade it. It may also inform a means to appropriately regulate user actions by adopting their own conscious standards of behavior, instead of enforcing it through strict legal regulations (Hsieh & Yeh, 2012).

Not all members of the organization come from similar backgrounds and levels of experience. Some members may be more willing to search for software solutions outside the scope of what they are provided, while others are more or less content with what they have been directly provided. Hsieh and Lee (2012) explored a particular relationship between a user's age and their views on USC. Their findings indicated that a user's age has a significant impact on whether or not they view USC as unethical. College-aged students in particular, were found to be much more tolerant of subverting guidelines and school policies regarding USC (Hsieh & Lee, 2012). This correlation has implications for

organizations like USMC, where in 2018, approximately 70 percent of active-duty enlisted Marines were between the ages of 17 and 24 (DOD, 2018). A large college-aged body of service members like this could share similar values in how they ethically view USC and other means of subverting organizational guidelines related to unauthorized software use. Security, education, training, and awareness (SETA) programs already exist as requisite courses for Marines to complete on an annual basis and have shown to be capable means of improving security posture through awareness of information system risks (Haeussinger & Kranz, 2013). However, in the face of completing a requisite task versus maintaining security compliance, Marines may find the ethical line between choosing one option over the other unclear, especially in the line of work that many find themselves in.

Regarding leadership's role in the ethics of information security and unauthorized software use, there are conflicting studies on its overall efficacy. For example, Xue and associates (2018) hypothesized that ethical leaders who reward and punish employees' security behavior would positively impact the overall information security climate. In contrast, Kumar et al. (2020) found ethical leadership behavior to be one of the least essential factors among senior management's effectiveness in promoting strong security behaviors. Instead, their commitment to enhancing security postures and dissuading behavior was a much more effective factor (Kumar et al., 2020). This is not to say that setting a strong ethical example is not important; however, proactive measures on behalf of organizational leaders to prevent unauthorized software appear to be an essential method.

2. Distributor Considerations

In addition to the user base, organizations must also be cognizant of the actions of the distributors themselves who provide third-party software solutions. Take Microsoft, for example, a high-profile company that has become a household name throughout many countries. When users began to continue to utilize versions of Windows that were out of date and had expired licenses, Microsoft continued to push security-related patches to them (Lahiri, 2011). This is an important practice regarding security concerns; however,

the behavior incentivizes users to continue the practice of using unlicensed software and circumvent Digital Rights Management (DRM) where possible (Lahiri, 2011). A large enterprise that relies heavily on third-party software to conduct standard business functions could find itself in a predicament where much of that software is being operated on an expired license. Users nonetheless will access it as the distributor has not enabled a means to prevent them from doing so. Unless the license contract has stated otherwise, this could cost an organization financially if they are found liable for a breach of a software license contract.

Gao (2022) categorized distributors and vendors into two separate categories, those that are proprietary and those that are open-source. While both categories compete for a higher share of the market, Gao (2022) concluded that existing risks of security threats benefit each at the detriment of the user, as it softens price competition. This further reinforces the notion that customers, even those as large as the USMC or DOD, should not place significant reliance on behalf of virtually any vendor to ensure that contracted software is secure. Even in the early stages of product deployment, vendors are financially incentivized to release a buggier product and simply patch the problem later in its life cycle, primarily due to the fixed cost nature of patching (Arora et al., 2006). These implications allude to the understanding that license management models, even at the enterprise level, should take caution when placing increased reliability on tracking and patching on the vendor's behalf.

C. SOFTWARE LICENSING

A comprehensive understanding of what a valid license entails is necessary to fully conceptualize the defined problem set of SLM. This includes the mechanisms put in place to validate a piece of software with a specific license and how they are handled between vendors and customers from a legal point of view. Though the use of software dominates how the modern world conducts business, the means to manage software licenses and contracts continue to be a complicated issue for many organizations.

1. Defining a License

At the core of this problem are two major aspects of software, tangibility and replication. Software is often contended as not tangible in the sense that it lacks physical form and characteristics and, as such, is associated as intellectual property (Phillips, 2009, p. xviii). One could argue the philosophical nature of the physical existence of software in the form of code or electronic signals, however, the abstract nature of these concepts will not be explored in this context. For purposes of this study, software is generally defined as a set of programs, procedures, and related documentation associated with a system (Phillips, 2009, p. 57). At the very least, it is easy to compare its tangibility to a piece of hardware, such as a laptop computer. When conducting an inventory of machines, the computer can physically be touched and inspected without a system to do so and absent of any electrical signals. On the other hand, software exists within the confines of a system, thus making the task of tracking it slightly more complex. As such, protecting intangible intellectual property like software becomes difficult when concerned with the possibility of its replication.

By its very nature, the same type of software is capable of existing on one or more systems. A single iteration is referred to as an instance of software, and many can exist at a given point in time. For example, a single server may retain the capacity of storing the source code for a particular program and is granted the ability to install instances of this program to any system that may request it. The server does not lose its instance, instead replicating the program to the new machine. This process can be repeated multiple times to create a limitless number of instances, making the problem set apparent. If intellectual property is replicated without authorization, there must be a means to prevent further use and distribution.

These fears of unauthorized distribution necessitate the requirement of a software license, the mechanism by which the licensor controls unauthorized utilization of its product and limits the licensee's rights (Kim, 2008). This is not to be confused with "selling" software, as this legally entails an entirely different set of rules regarding how the customer uses that software. When selling software, the licensor releases all possibility of control over its use and distribution (Kim, 2008). A license, on the other

hand, allows for the licensor to force the customer to accept a software license agreement, which, by extension, takes advantage of contract law and controls its use and distribution (Kim, 2008). A user may have access to an application to use it to conduct business; however, committing an act such as re-distributing its source code or using said application for illegal purposes may constitute a breach of contract. For an enterprise, such breaches can lead to severe legal and financial penalties that it may or not be able to fulfill.

2. Legal Considerations

There have been disagreements over whether one can consider software as actual property. Some believe that the software code itself is, in fact, the intellectual property of the licensor who in fact may do with it as they wish (Kim, 2008). Others challenge the notion that even if software is intellectual property, that still does not make it the same as "property," bringing into question whether the owner retains rights of exclusion (Kim, 2008). If they do not, then theoretically, a replicated instance of software may not be considered stolen, giving anyone other than the owner virtually unlimited access to it. However, this latter viewpoint appears not to be generally accepted by the software industry which has been capable of maintaining its right to control use through copyright law protections (Kim, 2008).

Examples of industries and firms incurring fees and penalties from illicit use of unlicensed software are numerous and can be financially devasting to offenders and software businesses alike. As far back as 2004, the Business Software Alliance (BSA), a trade group representing commercial software developers, fined Consolidated Engineering Services (CES) for its utilization of unlicensed software (Hughlett, 2004). Products from Adobe, Autodesk, McAfee, and Symantec were included in this fine, totaling $77,644 (Hughlett, 2004). Another example involves a labeling company based in the United Kingdom, which was forced to pay £24,800 for the use of unlicensed Microsoft software (Hall, 2011). Such fines may appear negligible for an enterprise encompassing over 100,000 employees, such as the USMC, however, for smaller organizations they can make or break their ability to keep their businesses afloat.

Moreover, should the BSA prove in court that an organization has willfully infringed software copyright law, the fine can reach as high as $150,000 per product, including lawyer fees (Bates, 2017). Not all licenses are the same, however, and distinguishing whether or not one is being utilized illegally is aided by understanding each of the different types.

3. License Models

A thorough search via open-source methods will uncover a multitude of license models that are recognized by legal experts, governments, and businesses alike. Five models in particular are outlined for this research. Figure 3 provides a graphical depiction by the Free Software Foundation (FSF) showing how these five models relate to one another and where they may overlap from a legal standpoint. The terms Free and Open Source (FOSS) can be seen in this diagram and refers to all software that also incorporates its source code as available to other parties. The following five descriptions are paraphrased from Synopsys Inc., an electronic design company that focuses on silicon intellectual property, verification, and software security (Synopsys Inc., 2020).

- Public Domain: Software within the public domain is considered the most permissive. Anyone has the authority to copy, modify, and distribute without restriction. This does not necessarily mean that it is safe from a cybersecurity standpoint, however legally, it is open and free to utilize.

- Permissive: These are also known as "Apache-style" and are also the most popular license type. They possess a few minimal requirements on how the software can be utilized, modified, and redistributed.

- LGPL: The Lesser General Public License (LGPL) allows for software to link other open-source libraries within its own code. One may even release their own application with such links under another licensing model. However, should any portions of the library be modified or copied into the code, then that software is restricted to being safeguarded only under LGPL terms.

- Copyleft: Also known as reciprocal or restrictive licenses, allows for the modification of licensed code and its redistribution. However, these new works must retain the same software license type as they originate from.

- Proprietary: This is the most restrictive form of license. All rights are reserved for the respective owner of the software, and works are not allowed to be modified or redistributed under any circumstances without the owner's express approval.

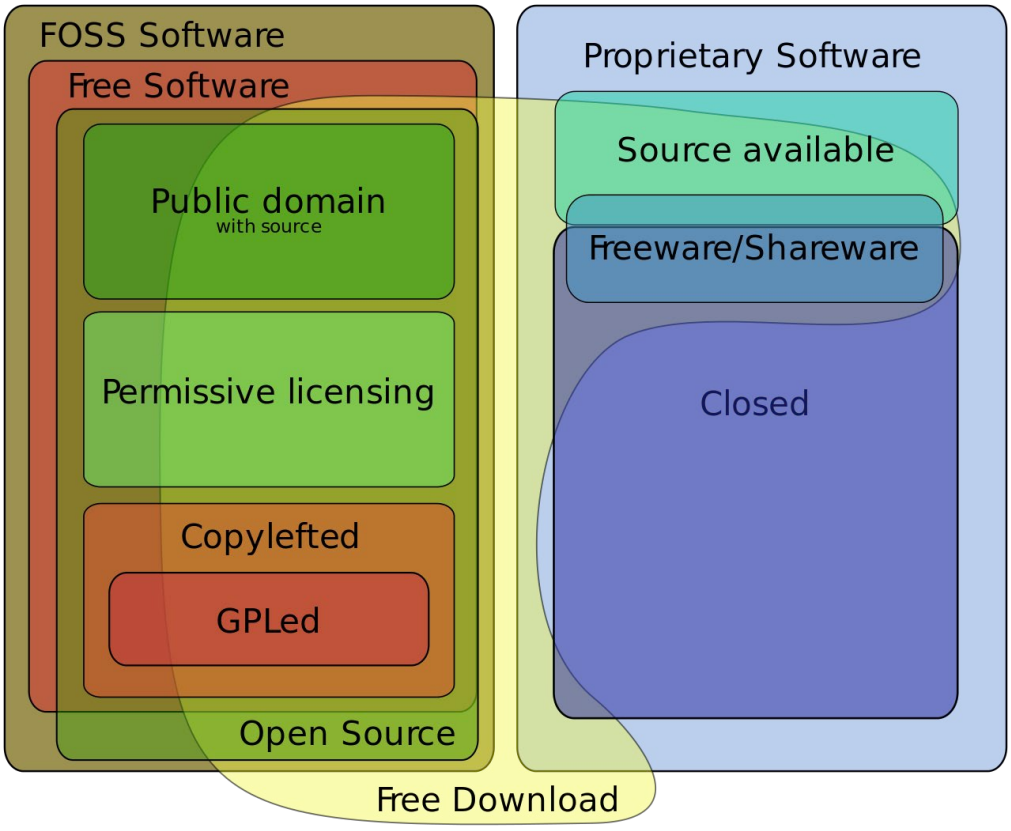

Figure 3. Graphical Depiction of Software License Model Relationships Based on FSF Free Software Definition. Source: The Free Software Foundation Inc. (2016).

The benefits of offering software solutions to the public free of charge are not readily apparent. Releasing a viable software solution that takes significant time, effort, and money to develop but never reaps any financial benefit upon delivery seems untenable as a business model. In reality, the business model itself appears to be the most

influential factor in determining whether or not software will go open-source over proprietary (Lindman et al., 2010). For example, some businesses will opt for a more service-oriented model, offering support and consultation work at a cost while avoiding license-based pricing (Lindman, 2010). However, a more recent study has placed software dependencies as a much more significant contributing factor to the choice of an OS licensing model, which impose restrictions on the part of the vendor (Liu et al., 2021). In brief, a developer's dependency on utilizing other third-party libraries and GPLs for its own projects prevents it from having freedom of choice in selecting a licensing option other than open-source (Liu et al., 2021). Recognizing how and why vendors and developers choose one licensing model over another may factor into how an enterprise manages and tracks those licenses. A license, for example, that is free to download and free of charge may come with its own limitations regarding how it is supported, and subordinate organizations that are caught up in unnecessary support contracts may end up financially costing the enterprise severely in the long run.

D. ENTERPRISE SOFTWARE MANAGEMENT

From a management perspective, monitoring all instances of software that are currently stored or running on each of the organization's systems is a discouraging endeavor. Unlike hardware, conducting inventory on software cannot be accomplished through a simple count of tangible assets. A sophisticated method of discovering and tracking instances is then required to accomplish what a hand count cannot. Furthermore, computer assets are likely to be dispersed across a wide geographical region, especially those that fall under the ownership of the DOD. Understanding what options have been explored and are currently available to accomplish this management feat is critical to resolving the defined problem set.

1. SLM Methodology

Due to the expansive number of available license types available to the world, it should come as no surprise that a suite of methods to manage them comes with. These may range from technology-based practices that restrict a user's ability to access software without a license to establishing written policies and governance that do no more than

define what is and is not allowed regarding SLM. The following list details many of the most common methods of SLM that are seen and practiced today (Watts & Davis, 2018):

- Per User: Licenses are distributed to specific individuals who have the authority to utilize them across one or more different devices.
- Per Device: Only one machine may be allowed to operate a specified license.
- Per Network: All machines that exist on an authorized network are allowed to utilize software based on the specified license.
- Subscription Based: Licenses by subscription allow a user or machine to operate software within a specified time frame. This time frame may be extended as the customer continues to pay the software owner to continue use.
- Management via Database: License and contract information is stored in a specially configured database, which may include many servers or cores. This information is then reconciled when compared to actual software use throughout the enterprise. This particular method can be costly and complicated to execute but is often necessary for larger enterprises.

As open-source software (OSS) has become increasingly popular in recent years, additional methods and approaches to management are sought after due to their high level of availability and distribution (Kumar et al., 2022). Violations of terms of conditions may occur with redistribution of modified OSS on behalf of a contributor or third-party user, with no effective means of enforcement through the tracking violator identified through digital footprints (Kumar et al., 2022). Figure 4 displays a method to counter these problems by utilizing blockchain components and smart contracts; if successful, this utilization could make breaching of licensing terms much easier to identify and enforce. Users violating OSS terms and conditions on workspace machines may inadvertently place the organization itself responsible. Advancements in enforcement could lead to great financial risk through litigations and settlements resulting from these violations, further necessitating effective SLM methods and policies.

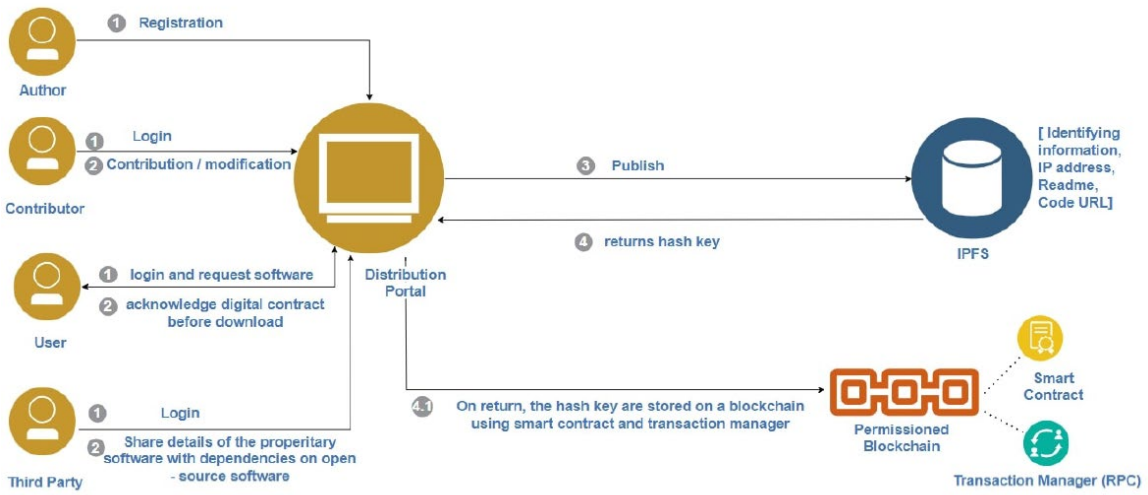

Figure 4. OSS Stakeholder Interactions Utilizing Blockchains, Smart Contracts, and InterPlanetary File Systems (IPFS). Source: Kumar et al. (2022).

2. Network Scanning and Software Discovery

A concurrent process with the reconciliation of valid licenses across the network determines what applications actively run on the network. Without the ability to stay up to date on what exists in a dynamic software environment, network administrators will find it very difficult if not impossible, to manage or prevent intrusion and exposure of their systems. Automated software discovery tools can assist with this endeavor and BMC, Tanium, Microsoft, and BelManage are among the companies offering solutions that assist enterprises with collecting and organizing software discovery data. However, there is no one-size-fits-all solution to software discovery, especially in organizations whose networks are numerous and span multiple domains across the entire globe.

One of the necessities for network scanning is the existence of software vulnerabilities, which can be labeled as specific flaws or oversights in the software code itself that can enable an attacker to conduct malicious activity through it (Dowd et al., 2007). While on the surface this is a technical problem, there is no mistake that this issue is the manifestation of human error (Ghaffarian & Shahriari, 2018). This inherently assumes that absolute trust cannot be reasonably established for the security compliance of any particular piece of software, requiring network scanning to become a near constant

requirement for the enterprise (Ghaffarian & Shahriari, 2018). Conventional scanning approaches include static, dynamic, and hybrid approaches to anomaly detection; these approaches vary by the necessity of conducting a software scan by actively executing its source code (Ghaffarian & Shahriari, 2018). However, a scan at this point assumes that software has already passed information assurance requirements by the organization and is conducted as a proactive measure against potential vulnerabilities. Beyond cyber security, the software discovery process serves a much larger purpose by not just determining the security compliance of what exists but what exists.

The growing necessity of maintaining an up-to-date and complete picture of an enterprise network is nothing new to enterprise IT. Identifying the components and applications on the network, how they are configured, and how they relate and communicate to each other is critical to analyzing and implementing any necessary changes that optimize security and functionality (Binz et al., 2013). One of the fundamental approaches to tackling this problem is the utilization of an Enterprise Topology Graph (ETG), which provides a technical snapshot of the services, processes, software, and infrastructure and their related dependencies on the network (Binz et al., 2013). Figure 5 shows a simple version of the systems that an ETG encompasses and how those systems are split among three layers: Graphical User Interface (GUI), means of discovery, and the data itself (Binz et al., 2013).

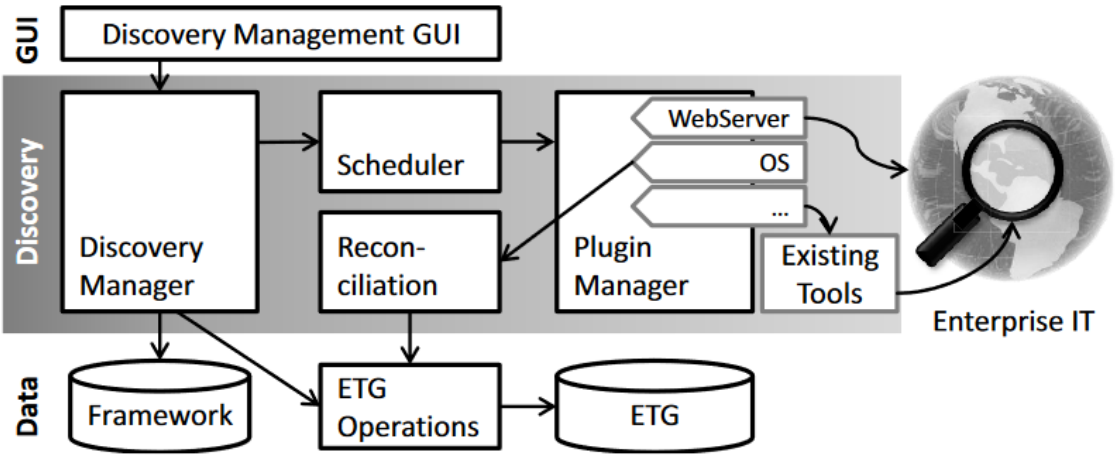

Figure 5. Enterprise Topology Graph. Source: Binz et al. (2013).

Automation continues to be one of the most challenging software discovery tasks for an enterprise. The number of applications running on an enterprise architecture may be so vast that the idea of each application pushing its own information up to a discovery manager or configuration management database (CMDB) is practically unfeasible. Too many client machines pushing updated information may utilize more bandwidth than the network can handle, hindering operational capacity as users wait for their machines to finish sending updates. Instead, plugins are developed to pull this information from the network and its applications and in such a way to ensure the application need not be aware that its information is being pulled (Binz et al., 2013). These plugins are, in essence, very complex in that they must accomplish the task of finding software, extracting information, modifying that information into an understandable language, and storing it in a database, all while minimizing operational impact to the user (Binz et al., 2013). This is achievable across a single network as all plugins will only have the parameters and configurations of that network to adhere to. However, a much more complex automation system is required across multiple networks and domains, especially those that are temporary or standalone.

E. SUMMARY

Enterprise IT management teams find themselves balancing a multitude of requirements to manage the primary success enablers of their organizations. They must fulfill the capabilities requested of their users to ensure operational functions are achieved while also safeguarding the very information those users create and process. Furthermore, each of these actions is constrained by the legal contracts surrounding the software used to conduct these functions. These software licenses vary in the methods in which they are managed, and the enterprise must be careful not to breach license agreements as set forth by the software owner or face legal consequences. Users may seek solutions to their operational requirements that breach these agreements, and therefore dedicated software discovery methods are necessary to highlight and respond to these breaches. However, the larger the enterprise, the more complicated SLM becomes, and the prospect of applying a single solution becomes no longer viable.

THIS PAGE INTENTIONALLY LEFT BLANK

III. RESEARCH METHODS

A. DESIGN AND PARTICIPANTS

1. Explorative Design

A qualitative research design was utilized to understand better how each research question is approached at a higher organizational level. Comprehending what methods and practices currently exist for enterprise software license management and software discovery is critical to establishing a baseline to apply to future solutions. More importantly, how one organization approaches these issues may be sufficient within its own context and business practices, but that does not mean that those practices apply to USMC enterprise networks. Comparing responses from each organization and their operational traits aided in determining what can and cannot be adopted in a military setting.

Expert opinion on this subject from a management perspective was considered a critical component of this research; therefore, conducting a series of interviews was deemed the most appropriate method. By responding to several open and closed-ended questions, interviewees were allowed to provide insight about the problem set that only their own experiences can allow. Discovering a definitive solution that solves the challenge of SLM and software discovery across multiple networks is not necessarily unattainable. However, approaching these issues with this definitive goal detracts from one's ability to holistically understand its complexities. It was deemed necessary then through the interview process to allow individuals to explore problems they faced within their own organizations and offer their potential solutions. Compiling this information enabled an analysis of organizational trends and anecdotal instances that went beyond seeking an answer and instead helped uncover underlying complexities within the problem set.

2. Criteria for Interviewed Organizations

Organizations selected for interviews needed to maintain several similarities to the USMC enterprise to receive and analyze relevant data. Approaches to SLM, software discovery and unauthorized use of software were expected to vary considerably, depending on numerous variables such as size, business function, geographical location, or even governmental affiliation. While not meeting each criterion was required, organizations that met similar attributes to the USMC enterprise were considered to have more relevant and applicable solutions.

- Government affiliation: Organizations were preferred to maintain some form of affiliation with the United States government. As a result, they are beholden to specific laws and functions that do not always apply to privately owned ones.
- Military affiliation: In addition to the rules and regulations that government organizations must follow, military organizations are required to follow even stricter policies for national security purposes. As such, interviewed organizations that adhered to the same or similar security policies were vastly preferred over those that can ignore them.
- Size: Larger organizations with over 1,000 employees or users were preferred over those that are much smaller. Those that are too small may not experience SLM challenges that effectively mirror those that currently face the USMC enterprise; thus, solutions they may provide were not expected to be applicable.
- Geographical Dispersion: The USMC operates continuously in areas across the entire globe. This entails maintaining several complex networked systems and physically and logically separate domains.

3. Selected Organizations

The following organizations were selected and either participated in the interview process or completed an online questionnaire with the same questions pool. For purposes

of confidentiality and security, names of respondents are not provided and specified responses are not tied to a particular organization. All organizations are listed in no particular order.

- Program Executive Office Command Control Communications Tactical (PEO C3T) - Mission Command Support Center, U.S. Army
- National Aeronautics and Space Administration (NASA) Johnson Space Center
- Marine Corps Systems Command (MCSC), USMC
- Marine Corps Installations East (MCIEAST), USMC
- NPS Information Technology and Communications Services (ITACS)
- Naval Cyber Defense Operations Command (NCDOC), U.S. Navy

B. MATERIALS

1. Means of Collection

Two methods were utilized to collect data from selected respondents. The primary method was to conduct in-person or virtual interviews, allowing respondents the opportunity to elaborate on their responses and ensure that those responses were based on their own experience and expertise. Virtual interviews were enabled utilizing the Microsoft Teams application, which allowed them to be both recorded and transcribed with the respondents' permission. In-person interviews were similarly recorded with permission; however, no transcription tool was utilized to log conversations. While most interviews were conducted through a one-on-one session with just the interviewer and interviewee, there were instances where multiple respondents were present for a single interview. In these cases, all respondents were given the opportunity to respond to each question, though in some cases, only one response was given by a single individual. All interviews were conducted at the unclassified level.

If an individual was not capable of conducting an in-person or virtual interview, the option to complete a questionnaire was made available. This was

accomplished using the Qualtrics XM questionnaire portal provided by NPS ITACS. Upon request, a link to the questionnaire was submitted to respondents Data collected from this method was less desired, as it did not provide interviewers an opportunity to request elaboration on responses where appropriate. For any questions that did allow for lengthened responses, the questionnaire allowed for ample space for respondents to provide as much detail and explanation as needed to justify their responses. Any data collected from this method was restricted from distribution outside those involved with the study to protect identities and confidentiality of respondents.

2. Interview Questionnaire

Interviewees were asked to respond to a total of approximately twenty-nine questions. This questionnaire was submitted to the NPS Institutional Review Board prior to any scheduled interview taking place, and was determined to not involve any human subjects research. Depending on certain responses, some questions were omitted per conditions of earlier questions. General information about the organization was initially requested to provide the interviewer with an understanding of the organization's functions as well as the particular position that the interviewee(s) held within it. Additionally, the interviewees' perceptions of their IT network architecture were requested to determine a general understanding of how it is structured. Personally identifiable information (PII) is withheld in this research to preserve the identity of those interviewed.

Upon providing general organizational information, the remaining questions focused on three separate but related topics of interest that focus on each research question. This began by generating an understanding of how the interviewee interpreted the issue of utilizing unauthorized software within the context of their own organization. For example, one question asked whether users may install third-party software onto their workspace computers without prior authorization. Others were more subjective, inquiring if the interviewee noticed any trends that indicated whether a user is more likely to use software not directly provided to them by the organization for work purposes.

The remaining questions focused specifically on the organization's ability to handle SLM and its capability to conduct software discovery using manual or automatic methods. A baseline was established to determine how reliant interviewees perceived their organizations were on third-party software to conduct business functions. Following that, they were asked about how they managed the licenses of third-party software, whether that was considered challenging, and how it managed to determine what software was actually running on their networks at any given time. With a few exceptions, respondents were allowed to elaborate on their answers as much as possible to each question. This was allowed to help provide deeper insight into the problem set, and allow the respondent to provide sufficient context and justification. A comprehensive list of each interview question is provided in Appendix A.

3. Modeling Tools

Multiple causal loop models were developed further to explore the necessity and results of this study. These models illustrate the relationship between different variables that exist within enterprise systems and aid in determining where change may be necessary for those systems to avoid undesirable outcomes. All models included in this thesis were designed utilizing the Vensim® modeling software tools developed and copyrighted by Ventana Systems, Inc. All terms of the license agreement for this software were adhered to in the development and publishing of this research.

C. PROCEDURE

1. Conduct of Interviews

Each interview was conducted with the intention of allowing respondents to provide as much information as they desired and as permitted by classification limitations. All parties understood that the process would take approximately one hour to complete, and interviewees had the freedom to halt the interview at any time. Additionally, any questions that respondents felt uncomfortable responding to were permitted to be skipped. These reasons included but were not limited to security restrictions, lack of expertise on the subject, or preference to defer the answer to another

subject matter expert. Negative responses also aided in identifying potential knowledge gaps that organizations may have regarding any of the three major research topics.

While several questions were limited to singular responses, most offered the respondents as much time and justification as desired to provide a sufficient answer. The interviewer ensured that they did not interject or interrupt during a response and provided clarification as needed in the event any question was not properly understood. On multiple occasions, topics of discussion deviated from the initial proposed question. This was considered acceptable and desired throughout each interview, as any additional insight into SLM, software discovery, or prevention of unauthorized use of software could be captured and analyzed as a critical data point. In the event that the discussion shifted too far from the research subjects in question, the interviewer was responsible for getting participants back on topic.

At the conclusion of each interview, respondents were allowed to ask any additional questions or provide any additional information they felt necessary. This information would be included as data to be analyzed as appropriate. Recordings and transcriptions were then officially logged and saved through Microsoft Teams, where applicable for virtual interviews. Respondents were not provided copies of either the recording or transcription from the interviewer, however, they would be made available upon request. Follow-on interviews were not required as part of this research.

2. Analysis Method

Utilizing a mixed method of extracting text from transcriptions in Microsoft Teams and clarifying that text from recorded audio, responses were then logged in table format. This format provided a single snapshot of all questions listed along a Y-axis while providing each organization its own column designated for responses. Placing responses side-by-side in this manner enabled the researcher to compare and contrast varying responses to the same question legibly, making notable trends and anecdotal variations easier to recognize and capture. Any information from recordings or transcriptions not considered useful or relevant to the nature of the study was not included in the table of responses.

In addition to logging the responses to each interview and questionnaire, an additional column was included in the response table for each organization. This column contained an interpretation of all responses on behalf of the researcher. Including this information served two separate purposes for this research. First, it provided the researcher with a means by which to record and share their own thoughts, interpretations, and insights from each response to each question from each organization. Second, it allowed for a way to determine the relevancy of each response to the original problem set as defined by MCSC. In the event an organization had provided a solution to a problem as complex as SLM, it could not be taken as the fact that this same solution could be appropriately applied to the USMC enterprise. An interpretation of these solutions needed to be applied by cross-referencing variables such as organizational size, military affiliation, and the number of networked domains that the organization operates with.

The responses presented multiple software tools and programs as potential solutions to assist with SLM and software discovery. In order to obtain additional information about these tools, open-source methods of collecting information about them through literature and official media were utilized. Biases inherent to companies' self-promoting software solutions were considered when investigating the capabilities and limitations of these programs.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. RESULTS AND ANALYSIS

A. MODELING THE PROBLEM SPACE

Before examining each research question, it is critical to explore how these problems are potentially connected and interrelated. Reviewing the concepts of unauthorized software use, licensing, and discovery in the context of enterprise software management is crucial to its merit. However, selecting and applying solutions for one issue may inadvertently introduce solutions for each of the others or perhaps introduce a separate but previously unknown or ignored problem. The following models assisted in identifying where some of these connections may exist, such that anticipated solutions would not be examined in a vacuum.

1. Underlying Issues

Additional behavioral patterns, supporting structures, and mental models are identified as potential variables that underlie the surface issues utilizing the systems thinking tool of the iceberg model (Goodman, 2002). While many interview respondents were not asked to directly discuss these possible underlying causes, they were highlighted in many interview responses. Figure 6 depicts an iceberg model related to topics identified as surface-level issues through each research question. Various patterns of behavior and potential influences on those surface problems considered present are seen below the water level. No causal determination has been made to determine how these unseen problems affect the surface; however, potential solutions that fail to address them may inadvertently waste resources when those visible problems eventually resurface.

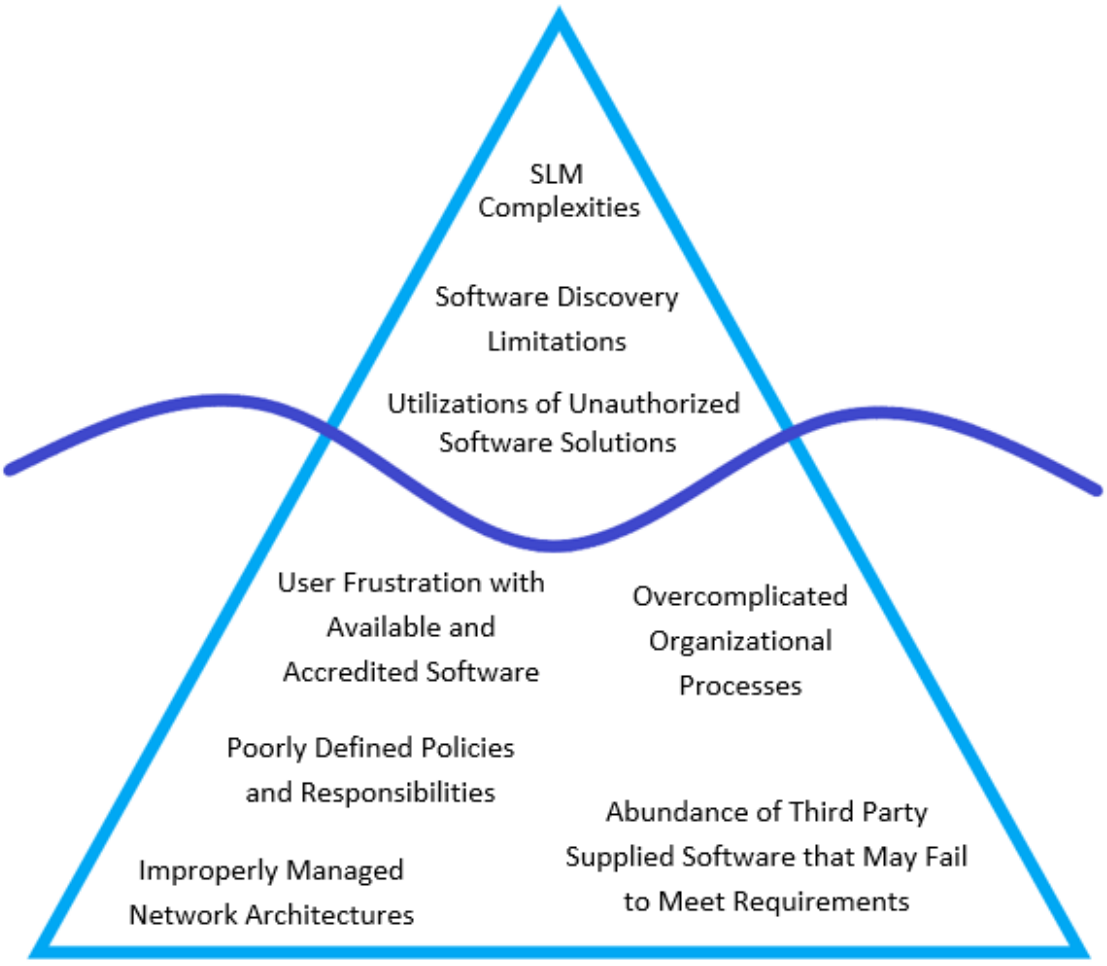

Figure 6. Iceberg Model. Adapted from Goodman (2002).

Within this model, the layering of each event, pattern, or behavior is not dictated by the direct cause and effect of the ones above and below each instance. Rather, the focus is on determining what is seen versus what is unseen, with the level of depth demonstrating increasing loss in visibility. For example, it was readily apparent in every interview that SLM is a complicated endeavor for an enterprise to control. However, the ability to discern how organizational policies may be impacting that surface-level problem was much more difficult. While some respondents were able to identify and even provide a source for policies that dictate why they manage their networks the way they do, most could not elaborate in detail on any of their organization's policies

regarding SLM or governance of unauthorized software utilization. This should not come as surprising as the official written policy is often, by design, complicated, lengthy, and drafted as a document for legal purposes and reference rather than recollection. Regardless, failing to look back at policy and determine whether it acts as a barrier to forward progress will ultimately halt any meaningful positive change in the organization. If a solution is adequate but not legal, even for unwarranted reasons, then it cannot be considered a solution until deeper change occurs.

2. Shifting the Burden

In his book *The Fifth Disciple*, Peter Senge (1990) identifies numerous system archetypes that can be used to identify and correct systems problems. The archetype, shifting the burden, describes a scenario where a short-term solution is applied to correct a problem, showing immediate results but, over time, atrophying the possibility for a fundamental solution (Senge, 1990). In the context of SLM, this archetype can be applied to emphasize the impacts of heavy reliance on technical solutions to the problem. Figure 7 displays a shifting the burden archetype in this context.

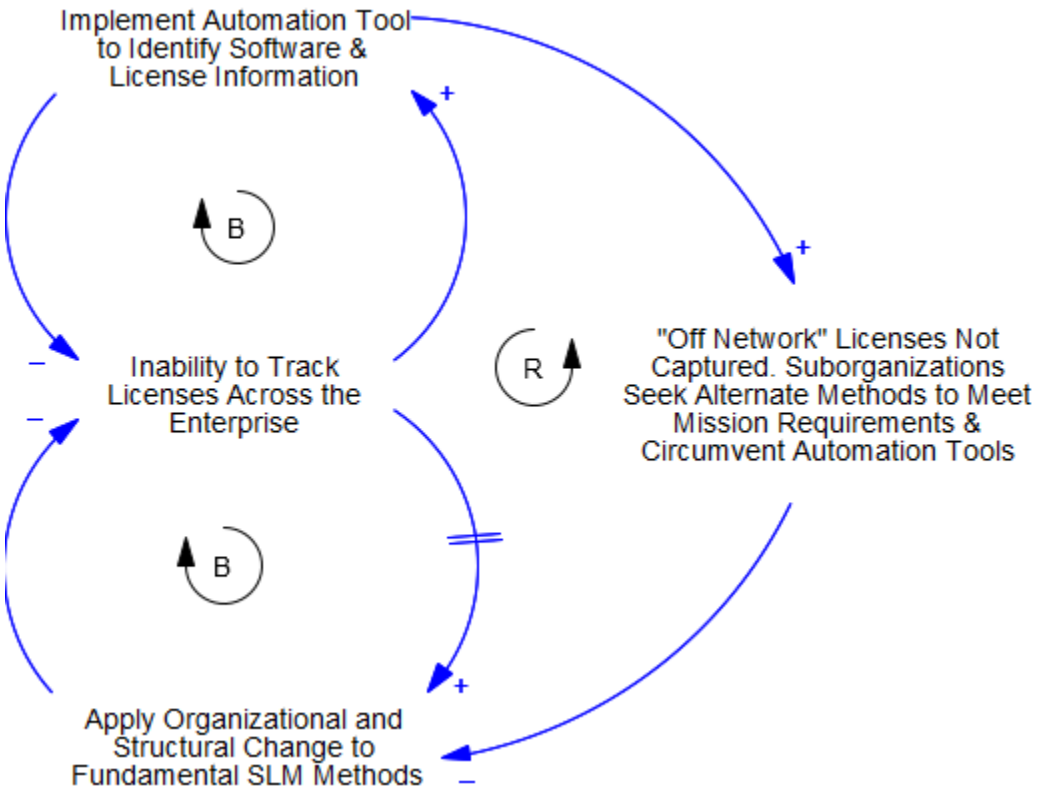

Figure 7. "Shifting the Burden": Utilizing Automated Tools for SLM.

This example demonstrates the potential outcome of applying an unnecessary amount of reliance on automation tools to scan and reconcile software license discrepancies on one or more networks. Theoretically, a suite of software tools could scan all workstations within its network to determine what other software is running on it and whether the licenses for those instances are up to date and assigned to the appropriate users. This makes managing these licenses faster and easier, but when relied on by itself to correct the problem, it may lead to further issues. For example, scanning a user's machine may have an unintended consequence of slowing down productivity as the machine utilizes processing power to submit scan results to the server where the automated scanning tool resides. Additionally, this tool is incapable of conducting any scans on networks and domains that possess no external network connectivity. In the case of USMC, these networks may include exercise or testing networks utilized by various commands for their own particular mission sets.

Consequently, individuals and unit commands will recognize the benefit of utilizing off-network machines to operate unlicensed software. The fact that this does not legally make it permissible to use this software is irrelevant to these users, as their priorities lie with accomplishing their mission requirements above all else. By neglecting to hold these commands accountable or perhaps providing them an incentive to utilize valid licenses, the problem symptom of SLM continues to persist. Furthermore, the application of the fundamental solution is ultimately delayed in its implementation due to the number of resources applied to the symptomatic solution. This is not to say that the symptomatic solution of using automated tracking software should be entirely avoided. Instead, it highlights how a hyper-focused approach to a promising solution may lead to delaying the desired end state.

3. Fixes that Fail

While shifting the burden is problematic due to costly delays, the "fixes that fail" system archetype is potentially more disastrous for an organization. In this model, a supposed quick fix to a problem is effective in the short term but leads to unforeseen consequences that continuously require the same fix to be applied until it becomes no longer sustainable (Senge, 1990). Interview data indicated a significant reliance on manual means to reconcile discrepancies in active software licenses versus actual deployed and active instances of the related software. These findings will be explored in later sections of this study; however, they provide a unique opportunity to demonstrate the impact of quick fixes to complicated solutions. A balancing loop of the manual quick fix and its associated problem can be seen in Figure 8, along with the reinforcing loop with the unintended consequence that can further degrade the organization's ability to manage licenses.

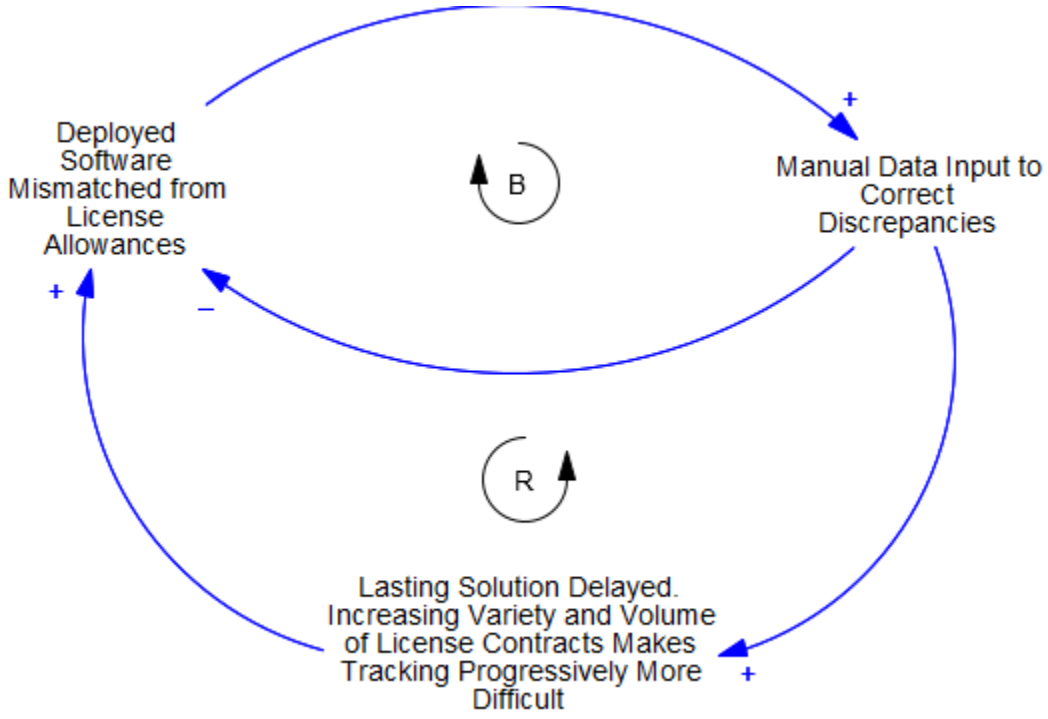

Figure 8. "Fixes That Fail": Manually Correcting Licensing Data Discrepancies.

In this example archetype, the continuous application of manual labor is not inherently leading to eventual failure in the organization. The fix may be costly and time consuming, but it nonetheless can complete the necessary task of license reconciliation. The danger in its lead up to eventual failure is through the potential rise in the variety and volume of licensing contracts as the organizations becomes increasingly dependent on third-party software applications to conduct its mission. Vendors can provide built-in safeguards that alert users automatically when software expires, but this is still a niche luxury that cannot be accepted as a given. Manual consolidation of this data is prone to user error, and it is only a matter of time until the organization's acceptance of this risk leads to costly legal action as a result of a future error.

B. INTERVIEW QUESTIONNAIRE RESULTS

1. Factors Contributing to the Use of Unauthorized Software

To better understand the organizational policy regarding unauthorized software, respondents were questioned on whether their organization permits third-party software to be installed on networked machines without prior authorization. Not surprisingly, all but one did not allow it to occur, and none were unsure how their organization stood on this policy. Despite these responses, there existed some level of understanding that even though users were not permitted to do so, some still sought means to install third-party software without authorization or oversight. This level of uncertainty was not shared by all respondents however, as more than one viewed their level of security as capable of preventing this in virtually all instances. The single affirmative response to this in inquiry indicated that one organization did not see this as a threat to their security posture. Considering how this organization has also adopted bring-your-own-device (BYOD) as an option to conduct business, this also may simply be due to unavoidable circumstances. The degree to which military organizations prioritize the security of their information systems is undeniably high; thus, it is unlikely that BYOD will be adopted in the future, even when integrating through a virtual private network (VPN).

User propensity to use software that their organization did not previously authorize was further explored throughout each interview of this study. This behavior was considered critical to explore not only in the context of network security but also to determine how it may relate to license management. One of the reasons software licenses exist is to ensure that software use is employed by someone who has agreed to utilize it within the parameters set forth by the software owner. Suppose a significant number of users in an organization find means to operate unlicensed software on the organization's networks. In that case, it begins to bear undue legal, financial, and security risk, which call for SLM methods to counteract. An organization that identifies factors that increase this level of risk may find itself in a better position to take proactive measures to counter unauthorized use of software. Respondents were asked a multitude of questions aiming to highlight these potential factors, which are summarized as follows:

a. Users conducting project-based rather than functional duties

Respondents indicated that users whose work functions revolved around projectized or specialized requirements were much more likely to seek alternate software methods to complete their tasks. This was due to their rotating nature, which required consistent changes in methods to complete those tasks, which led to seeking different software applications to get the job done. Conversely, users who fulfilled purely functional roles that required them to complete the same tasks over a long period were less apt to deviate from software solutions provided by their organization. As long as what was provided allowed them to complete their daily requirements, there was no need to seek software solutions elsewhere.

b. Lack of web-based tools to operate business functions

One of the many benefits of keeping the internet accessible through a business network is the number of productivity tools. Web-based applications allow users to access and run processes that are often executed on platforms not located in the same domain as the user. Interview data indicated that so long as these tools are available through reliable internet access, users will prioritize these solutions over those that are executed directly on their workstations and not sanctioned by the organization. However, suppose an organization's network lacks access to the internet. It must consider the increased likelihood that users will introduce unauthorized software to accomplish tasks that they could otherwise accomplish through a Google search.

c. Availability to access freeware and open-source applications

Unlike web-based applications, open-source applications that are free to download are executed directly on the user's workstation. If an organization's policies do not prevent users from downloading these applications, it may find itself overwhelmed with FOSS installed on many of its assets. This is not inherently a negative outcome but brings with it the potential for the organization to become beholden and liable to the associated license agreements. Respondents indicated that users tend to view software that is free to download and utilize as a benefit, and therefore do not consider consequences when utilizing it commercially. For example, a software license may allow

a user to access its features for personal use, but once it is used as a tool to sell a commercial product, that software owner may now be entitled to compensation per the end-user license agreement (EULA).

d. Heavy reliance on soft policies to prevent installation rather than technology-enabled prevention methods

Most respondents claimed that their organizations put significant trust in technological solutions to prevent unauthorized software installation. This was opposed to adhering to written policy, which enables governance and liability over users' actions. Trusting any user, even those provided privileged access to the network, was deemed too high of a risk factor in nearly all interview responses. Some organizations are moving as far as adopting a zero-trust architecture (ZTA), which by default, must verify any device, system, user, or application regardless of its location on the network (Alevizos et al., 2022).

e. Access to stand-alone computer assets that can operate unlicensed software

Respondents indicated they were likely to seek off-network assets if a networked workstation did not allow users to install the desired application. Therefore, organizations that maintain stand-alone devices may be at risk for higher instances of unauthorized software utilization. This introduces no risk to the network from a security standpoint, as the device has no physical or logical connection that may introduce malware or security vulnerabilities to it. However, this still poses a potential problem when viewing this through the SLM perspective. Breaching a EULA is domain and network agnostic and may still impose legal and financial penalties depending on the contract within that EULA.

2. Approaches to SLM

Though each interviewed organization was identified as operating under the authority and direction of the U.S. government, there existed significant variances in their methods of handling SLM. This revelation clearly indicated that there is no direct law or policy governing how they must manage software licenses across their networks, so long

as those methods discourage fraud, waste, and abuse of taxpayer funding. Respondents were asked several questions regarding how their organization manages software licenses, both with technology-enabled tools and overarching policy. The following findings do not indicate methods shared among all respondents or their particular effectiveness, but rather provide a picture of what is currently in effect. In addition, no one method is utilized as a single solution. Each method retains the potential to work in tandem with others, providing organizations with a range of license management possibilities that synchronize best depending on the organization's needs.

a. Enterprise-wide license agreements

Arguably the most expensive and straightforward means of obtaining a license for the entire organization is one that authorizes all users to access the required software. This was most common in responses for products such as the Microsoft 10 operating system (OS) or Microsoft 365 suite of productivity applications, including Word, Excel, and PowerPoint. This is very efficient in mitigating risk across the organization, as the license is effectively cleared for use for a set number of users without the need to manage specific distribution methods. Respondents also indicated that this was still not entirely without risk, as there are often a specific number of licenses that are still tied to the contract agreement. For example, if an organization purchases an authorization to utilize 100,000 instances of a particular program, going over that instance by even one is a breach of contract, and the organization is liable for being over licensed.

Conversely, if the organization only utilizes 90% of those licenses across the enterprise, that additional 10% purchased becomes a sunk cost with no added benefit. It was clear throughout each interview that while this type of purchase agreement does occur, it is relatively unique to a select few software applications that can be determined as essential by every user across the organization. This makes this form of SLM useful in niche circumstances but impractical for most contracts across the enterprise.

b. Configuration Management Database (CMDB) Utilization

CMDBs are essentially files taking the form of a standardized database containing information relevant to the hardware and software components that are utilized across the

organization (Montgomery & Mixon, 2020). More than one respondent indicated the necessity of using a CMDB to assist with managing licenses. In the case of the USMC, they are one of the explicitly authorized methods to accomplish SLM. An example of a CMDB dashboard is displayed in Figure 9, which visualizes asset data in an easy-to-read format.

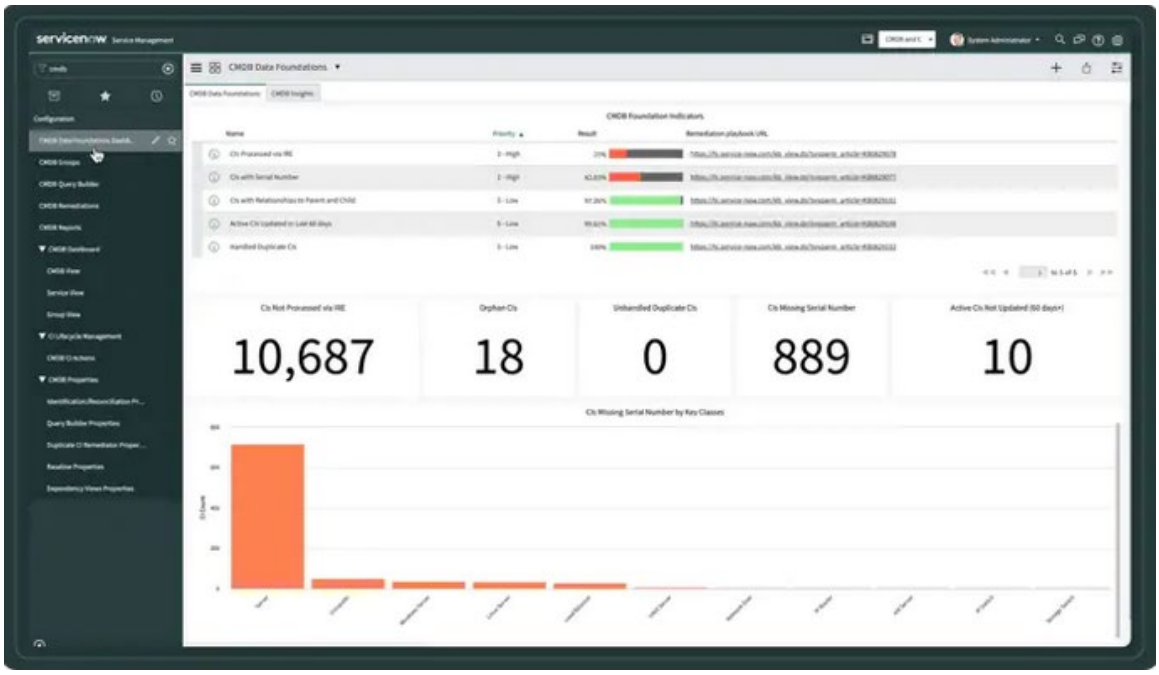

Figure 9. ServiceNow CMDB Dashboard Example. Source: ServiceNow Inc. (2022).

As a container, these were noted as effective tools that gave asset managers a comprehensive picture of what software exists on the network. However, nearly all respondents who utilized a CMDB highlighted their inability to automatically update information, particularly active software instances running on their network and their respective licenses. A database is only as effective as the information stored within it, and if those responsible for manually updating it are doing it incorrectly, it becomes a liability rather than an asset for the organization.

c. Licensing by workstation

Most respondents indicated preferences for assigning licenses to a particular workstation or computer over those assigned by the user. This method was done for particular instances of software that are necessary for some users to complete their daily work but were not considered needed for all users. In many interviews, a popular example of this was the Adobe Pro software suite, which allows users to edit portable data files (PDFs) in dynamic ways beyond just viewing and reading them (Adobe Inc., n.d.). Keeping these licenses grounded to a particular workstation was deemed to be an easier method to track licenses, as software is not required to jump from computer to computer in the event an authorized user changes workstations. However, this method had notable downsides, as respondents indicated situations in which multiple users would access a single machine that had a licensed instance of software. If the software license agreement indicated that only one user could access that software, then multiple users utilizing it on one machine would be considered a breach of contract. Some vendors are navigating this possibility by introducing multi-user licenses, which require users to log in with an ID before they can use licensed software.

d. Licensing by user

In contrast to licensing by workstation, several respondents stated that their organization's policy determined that licenses must only be assigned to specific users. From the user's standpoint, this method offers significant flexibility in that if they happen to switch positions or change workstations, they still retain the rights to utilize specific software. This was often done by assigning the user's account to a specific security group, which allows delivery of the desired software via the Microsoft System Center Configuration Manager (SCCM). Users may access a software center, similar to an application store, and download whatever program they need through this method. This process can become increasingly difficult to keep track of in a military setting where users come and go within billets every few years. In some cases, users may leave their billet altogether without conducting proper check-out procedures, leaving a paid license

potentially unused and unavailable. Without an explicit process to pass roles and licenses accordingly, organizations are left with virtual inventory that provides no value.

e. Vendor defined

One of the most significant patterns identified across all interviews was the increasing amount of control that vendors have maintained in the licensing process. When directly asked how their organization employs and distributes licenses, respondents would often indicate how that methodology is not always dependent on how they conduct business, for better or for worse. In some cases, this relinquishment of the burden was very much welcome. Having the vendor manage licenses through their web-based portal or user ID log-in system safeguards the organization from liability. If a program has built-in features that limit the possibility of someone using illegally, there are likely fewer instances of unauthorized use.

Conversely, having the entire process managed through vendor-defined means had its drawbacks for several organizations. For example, one respondent indicated how the CMDB was a necessary means to track licensing information by the policy. Oracle VMware's application comes with a proprietary license management portal, which project officers would utilize to manage them. Tracking licenses in the CMDB and the portal was considered duplicative work but was nonetheless required, which led to either the CMDB not being updated correctly or project officers wasting additional time doing the same job twice. Another respondent indicated that the fast-paced changes in technology influence numerous vendors to change their management processes at a cyclic rate. This becomes increasingly difficult to keep track of as government organizations become increasingly dependent on third-party software to perform essential business functions.

f. Manual tracking

Each respondent indicated at least some form of manual process that was critical to SLM. It appeared to be an unavoidable dilemma whether this labor involved data input to a CMDB, updating a Microsoft Excel spreadsheet that cross-referenced licenses with their users, or reconciling electronic file cabinets to retrieve or update contract

information. Roughly half of the respondents viewed this problem as a detriment to their capability to conduct SLM. As noted previously in this study, direct human input is inherently subject to error. Therefore, any process involved with SLM that is not automated will be subject to an undesirable level of risk. In this case, that risk may take the form of a license associated with the wrong user or updates to the database not occurring on time due to an asset manager not being available. Additionally, manual effort is subject to labor costs. For every aspect of SLM that a machine cannot complete, opportunity costs arise by pulling a qualified individual away from other duties to conduct it.

The remaining half of the respondents stated that while automation can have cost-saving benefits, tracking licenses via manual means is simply a requirement due to its complex nature. One respondent compared the process of SLM to that of project management. The life cycle of a third-party license contract, like a project, has a finite timeline in which it ends or requires renewal. Asset managers in this regard need only to monitor these timelines and engage with stakeholders when necessary to ensure the software product suite meets requirements and that the customers are operating within the bounds of that contract. In this regard, the processes this organization had in place were functional to the point that they did not merit a need to invest in automation.

g. Written policy and governance

Out of all means to implement and enforce SLM, the least trusted was that of relying on written policy. This is not to say that respondents stated that SLM policies are not necessary, as it was well understood that they serve as a fundamental foundation for governance. However, in terms of utilizing that policy as a direct means to enforce user activity, it was not deemed valuable. At best, they were deemed necessary to provide authorization and guidance on how licenses can and should be managed.

Previously, many respondents were unable to recall the titles and general contents of organizational policy as it relates to SLM but were able to identify where to reference it quickly. It would only be expected that if IT asset managers and network administrators cannot provide immediate recollection of policy details, which is natural, then the user

base should be believed to be even less capable. To this end, no organization should build a license management system based on trusting those users to "do the right thing."

h. Decentralization of responsibility

Out of all organizations interviewed, only one had taken the approach to delegate licensing responsibility to subordinate organizations. Through this approach, lower echelons were required to contract to license through their own methods without adhering to direct reporting to the larger organization as a whole. This approach was adopted to understand that each sub-organization retained a specific mission set that was unique to itself. Due to this level of uniqueness, only those who operated within the sub-organization were deemed knowledgeable enough to manage those software licenses necessary to accomplish those specific requirements. Through this method, overall policy and governance can still be applied. However, the larger organization was much more capable of focusing on enterprise-wide SLM as it was not required to manage and direct smaller networks. As a result of this dynamic, licensing was considered a challenge not because of the management aspect but acquisition. This was especially true throughout the COVID-19 pandemic, which forced many organizations to adopt software that satisfied increasing telework requirements hastily. Problems revolved much more around obtaining what users needed, such as VPN connectivity, rather than struggling with managing what was already in inventory.

3. Discovery and Management Tools

Numerous respondents often utilized commercially available software suites to assist in conducting network scanning, software discovery, and assist with SLM. The following were mentioned as being actively utilized by one or more respondents, with varying degrees of effectiveness. A deliberate universal needs statement (D-UNS) by MCSC identified several of these tools as already being unable to meet the desired capability of software discovery data across disparate networks (Toohey et al., 2020). The information provided by this D-UNS, combined with that of interview data, allows for a more comprehensive look at how these tools are utilized across multiple

organizations. All capabilities and limitations listed are restricted to the context of software discovery and SLM, and do not reflect the total utility of a particular system.

Table 1. SLM and Discovery Tools.

Tanium	Capabilities	- Real time scanning across visible networks - Is capable of integration with additional CMDB's such as those made by Salesforce or ServiceNow (Tanium Inc, 2022) - Retains historical data of previously scanned instances (Toohey et al., 2020)
	Limitations	- Not as effective at scanning systems on Linux OS (Toohey et al., 2020) - Client requires locally installed software, consuming storage space and processing capability when providing updates (Toohey et al., 2020) - Incapable of scanning non-endpoint devices (Toohey et al., 2020) - Cannot communicate discovery data to a CMDB - Appears to provide "false positives" by providing data that is not current along with data that is current
BMC Discovery	Capabilities	- Client machines do not require locally installed software to be scanned, however this feature is still available (Toohey et al., 2020) - Retains historical data of previously scanned instances (Toohey et al., 2020) - Provides multi-cloud level support (BMC Data Sheet, 2021) - Can be deployed as Software as a Service (SaaS), cloud, or on-premise (BMC Data Sheet, 2021) - Can store contract information that is used to buy software packages
	Limitations	- Incapable of scanning non-endpoint devices (Toohey et al., 2020) - Full discovery capabilities still require locally installed software (Toohey et al., 2020) - Cannot communicate discovery data to a CMDB
Big Fix	Capabilities	- Retains historical data of previously scanned instances (Toohey et al., 2020) - Can import security and compliance information to an external database
	Limitations	- Incapable of scanning non-endpoint devices (Toohey et al., 2020) - Focuses on security and vulnerabilities of platforms, does not provide information on licensing

BelManage	Capabilities	- Capable of storing discovery data into an external database - Domain agnostic, as long as client device has Bel Manage discovery software installed, it can pull discovery data from the client
	Limitations	- Incapable of scanning non-endpoint devices (Toohey et al., 2020) - Retains historical data of previously scanned instances (Toohey et al., 2020) - Client requires locally installed software, consuming storage space and processing capability when providing updates (Toohey et al., 2020)
ServiceNow CMDB	Capabilities	- Compatible with Bel Manage, Big Fix, and Tanium as a database back end - Enables viewing discovery data to visualize and audit discovered vs. purchased software - Can store contract entitlements to enable reconciliation of where license authorizations exist per user - Can update discovery data automatically when reconciling actual and should be software instances (Service Now Inc., 2021) - Discovers physical and logical infrastructure configuration items (CI) to include applications, containers, VMs, or storage devices (Service Now Inc., 2021)
	Limitations	- Automation capability appears to be difficult to integrate with other tools at the enterprise level
HBSS	Capabilities	- Retains historical data of previously scanned instances (Toohey et al., 2020) - Tracks software security vulnerabilities - Can prevent unwanted third-party software from being installed on client machines
	Limitations	- Client requires locally installed software, consuming storage space and processing capability when providing updates (Toohey et al., 2020) - Discovery data limited to endpoint instances
Microsoft Defender for Endpoint (MDE)	Capabilities	- Natively installed on Microsoft OS clients - Provides similar scanning and protection capabilities as HBSS - Retains a suite of tools focused on security, compliance, and access control
	Limitations	- Not useful on systems that do not run Windows OS - Discovery data limited to endpoint instances

Enterprise Mission Assurance Support Service (eMASS)	Capabilities	- DOD GOTS database that can track contracting and end user license information - Data consolidation assists not just with surveillance, but managing overall budget decisions - Assists with enforcing security compliance of client systems (Defense Information Systems Agency [DISA], n.d.) - Provided and supported directly by DISA (DISA, n.d.)
	Limitations	- Entire database appears to communicate with just itself, and may not provide information to other tools or systems (DISA, n.d.) - Suite of available tools better suited for security compliance rather than enterprise SLM
Microsoft SCCM (Renamed to Microsoft Endpoint Configuration Manager)	Capabilities	- Retains historical data of previously scanned instances (Toohey et al., 2020) - Can capture discovery data on endpoint and server assets (Toohey et al., 2020)
	Limitations	- Does not function with non-Windows OS devices (Toohey et al., 2020) - Client requires locally installed software, consuming storage space and processing capability when providing updates (Toohey et al., 2020) - Considered to be overall less functional than BelManage, which it replaced for the USMC (Toohey et al., 2020) - Cannot discover network transport devices (Toohey et al., 2020)

Out of all software packages mentioned, only the ServiceNow CMDB was stated as having the potential to meet an organization's needs for enterprise SLM. With this exception, all other cases involving discovery tools and external databases aired the grievance of manually updating database information with what the discovery tool was able to report. ServiceNow offers its proprietary discovery tools known as IT Operations Management (ITOM), that can communicate directly with the CMDB (ServiceNow Inc., 2021). This can discover physical and logical configuration items (CIs) such as virtual machines, containers, storage devices, and software applications on the network and the relationships between them (ServiceNow Inc., 2021). Purchasing duplicative software packages that can perform near similar discovery functionality however is not practical or cost effective, especially for those organizations that have already invested heavily in multiple discovery tools. Fortunately, ServiceNow CMDB also has the potential to

integrate directly with those tools already in use by the USMC, specifically Tanium and BMC Discovery (ServiceNow Inc., 2022). Given the potential that respondents have given this suite of services, it is highly recommended that the USMC explore ServiceNow as an option to meet its software discovery and SLM requirements.

C. COMPREHENSIVE ANALYSIS

1. Notable Trends

Throughout all interviews, multiple findings were shared among the majority of respondents and their organizations. These trends allude to challenges faced by government agencies and highlight areas of concern that may require future attention for change. Not all of these characteristics were shared by each respondent; however, the frequency in which they appeared make them worthwhile to consider.

a. Low degree of user trust

Most respondents, to some degree, discussed the importance of increasing user awareness regarding cyber security vulnerabilities and the dangers of utilizing and installing unauthorized software onto the organization's network. All interviews confirmed that they require cyber security training for their workforce annually. Despite requiring this training and increasing awareness, however, respondents made it very clear that their trust in the overall user base not to commit security violations was still very weak. Multiple organizations actively sought to adopt Zero Trust as a basis for the network security posture, which places a contingency on all users, even administrators, to have that trust earned and verified (Alevizos et al., 2022). Additionally, respondents brushed off many questions about users installing software of their volition as a non-issue. Workstations and their associated networks were often considered locked down. Even if users wanted to install third-party software, they would be entirely unable to do so without prior approval.

b. Perceptions of SLM as a persistent challenge

Whether an organization managed a single network on one site or multiple networks spanned across installations across the globe, all indicated that SLM was

considered a challenge. One of the primary considerations for these difficulties is managing a defined inventory that does not have any physical structure in the same vein as hardware. The concept of walking into a warehouse and verifying the tangible existence of a stock of items is impossible in this regard, but the existence of licensing contracts and agreements demand that some form of inventory must still occur to reduce risk. Other respondents noted that many of the challenges they faced today stemmed from the ever-changing nature of technology and how vendors choose to conduct business. For example, companies are trending towards subscription-based services instead of handing out a set number of license keys to be applied to a single user or workstation. This reduces overall risk by preventing unauthorized users from accessing software that they are not licensed to operate, but this forces changes within management systems at the enterprise level. Information within a CMDB may become no longer valid, and thousands of users could be forced to set up individual accounts for software access, all of which are now tracked entirely by the vendor.

One noticeable attribute that may be considered a variable in worsening this problem is centralizing management responsibility. Each organization that considered SLM a significant challenge was burdened with the responsibility of managing it through multiple sub-organizations. Respondents who claimed it was difficult in the acquisition and not management had taken steps to delegate SLM responsibility. This was deemed appropriate as smaller organizations are much more aware of their specific mission or business requirements than that of the larger enterprise. As long as those smaller organizations adhered to the overall SLM policy, there was no need to actively track and manage particular licenses on their behalf.

c. Consistent utilization of manual tracking for software licenses

As was previously indicated in the results of each interview, all respondents indicated a significant reliance on updating licensing information through a manual means. This varied from updating license contracts and entitlements within an electronic file cabinet to cross-checking a Microsoft Excel spreadsheet listing what users were entitled to operate specific software instances. In the problem statement of this study,

MCSC highlighted the reliance on manual input as one of the more significant problems that revolve around SLM, especially as it relates to lower reliability of data and higher cost. This continued to be a trend throughout all interviewed government organizations, highlighting two potential realities for this problem.

First, requiring manual input for a task in an environment as rapidly evolving as SLM is potentially unavoidable. The amount of control that organizations have over the ways in which vendors supply and design their software packages gives customers less flexibility in developing management methods. Even if the organization determines an optimal way to perform SLM at the enterprise level, one major vendor can change their license and distribution format to make them no longer compatible. To this degree, human interaction with the database that stores license contracts, keys, and entitlements may be inevitable. Second, the legal and contractual complexities around software licenses make it very difficult for a non-human system to manage at the enterprise level effectively. An allowable number of users, expiration dates, usage agreements, and distribution limitations are some of the many unique attributes of each license agreement. At the very least, each introduction of a new enterprise license agreement and its associated contract must be appropriately formatted to enable an automated system to properly manage that software's distribution, entitlements, and usage.

d. Third-party software reliance

Each respondent was questioned on how they believed their organization relied on third-party software to conduct major business functions. This question aimed to obtain a better understanding of government dependencies on outsourced software development and further highlight the importance of effective SLM. Unsurprisingly, all respondents stated that they heavily relied on third-party software to the point where in-house development rarely occurred, if at all. Even software that was considered government off the shelf (GOTS) was often expected to be developed utilizing commercial off the shelf (COTS) tools, making it not entirely GOTS. With technology advancing rapidly and with the advent of continuously specialized software development teams, these responses were not unexpected. They do, however, add weight to the consequences of not properly

managing licenses. When organizations lack the internal development capability to fill in requirement gaps, they increase their reliance on external developers to the point that losing it or failing to maintain it properly may lead to organizational dysfunction or collapse in the future.

This line of questioning led to an additional finding that was more unexpected, which was that of external development using this reliance to shape the requirements of government organizations. Product suites provided by Microsoft, Adobe, and Oracle were among the most common respondents indicated as being used most often by their users. When asked why such products were selected, respondents commonly stated that continued use of these software packages resulted from users becoming reliant on them throughout the years. If true, this reality alludes to the fact that private industry is growing more capable of defining government productivity requirements rather than by the organization's actual mission or business needs. Allowing outside agencies to shape these needs may force government organizations to spend more on software solutions that are not actually needed to complete their mission.

2. Functional System Model for Enterprise License Management

A causal loop model was developed from a system dynamics perspective to provide an overarching picture of how SLM can properly function in a large enterprise. This model considers user influence on purchasing software packages and how cost impacts software package retention, and introduces the concept of delegating SLM responsibility where appropriate.

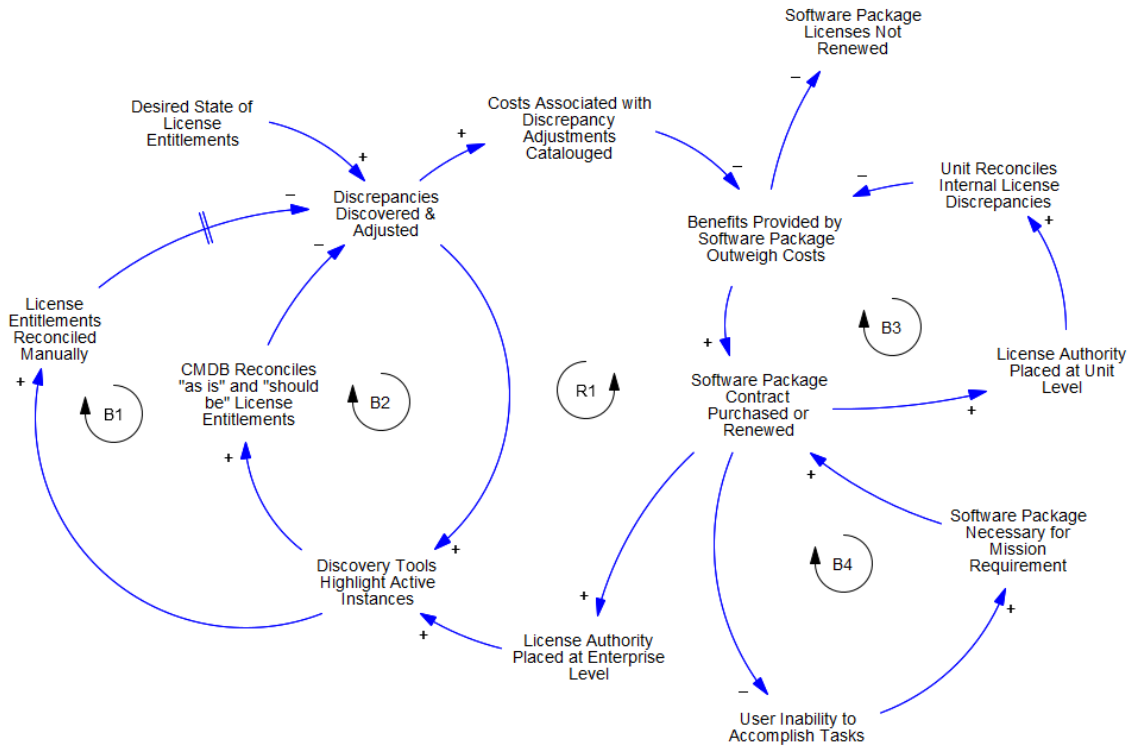

Figure 10. Enterprise License Management Systems Model.

One reinforcing loops and four balancing loops were identified within this model that helps regulate an enterprise's ability to provide the necessary software to its users while simultaneously reducing unnecessary cost. Each of these loops is explained in detail to help highlight their significance within the overall model.

B1. This is the currently in place "as-is" method of conducting SLM reconciliation manually. It can discover discrepancies within the system but is marked with a delay to signify the additional time and cost associated with not being capable of automating the process.

B2. With a properly configured CMDB capable of automatically reconciling discrepancies, the delay in correcting them is mitigated. However, it should be noted that automating the process does not fundamentally change the overall functionality or balance of the model itself. With the removal of B2, B1 still exists and can functionally still complete the necessary task of reconciliation, albeit much more slowly.

B3. All remaining packages that are still needed by sub-organizations but must be included on disparate or disconnected networks fall into this loop. These organizations retain SLM and software package life cycle responsibility. The larger enterprise still provides written policy to govern how this flow is conducted, but management methods and risk analysis is delegated to the sub-organization. This releases the burden of software discovery and discrepancy reconciliation for these licenses, especially when software discovery tools are incapable of conducting scans.

B4. The need for software in the first place is dictated by the user's needs, which drive their capacity to accomplish their associated mission or business requirements. As in place software packages fail, replacements are necessary, reinforcing the need to supply newer contracts. Once those contracts are fulfilled, users become capable of completing tasks that detract from the need for redundant solutions.

R1. The core system is represented through this loop, as it represents the life cycle of a software package being purchased through its eventual renewal or disposition. Specifically, this life cycle represents what the system should look like from the enterprise authority perspective. Software packages that are deemed not visible by the enterprise either due to where they are physically operated or the niche requirements of specific sub-organizations do not enter this loop. What makes this loop different from the others is that it is reinforcing, meaning that change is essentially compounding itself throughout the life cycles of multiple software packages (Senge, 1990). This represents the consistent need for additional software applications to meet business requirements despite some of those applications eventually being deemed unnecessary.

This model understandably represents only a portion of variables within a government IT enterprise. Capturing every intricate portion of the enterprise is exceptionally difficult due to the vast amount of moving parts, both human and non-human. Nevertheless, it achieves an executive-level view of how separate causal loops can effectively balance one another throughout the life cycle of a software package. IT Asset Management teams would do well to pay particular attention to those variables that exist at the epicenter of multiple causal loops; such is the case here with correcting license entitlement discrepancies and approving software package contracts for purchase

and implementation. Ensuring these processes function as desired will enable their adjacent and dependent functions to perform at similar levels.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION AND RECOMMENDATIONS

A. CONCLUSIONS

1. Summary of Findings and Assessments

The Marine Corps faces significant challenges with SLM, especially when tackling it at the highest echelons of the organization. Finding a means to automate the discovery and data consolidation process of hardware, software, and transport layer devices across multiple networks is cost-efficient but only hits at a component of the problem. While automation reduces time and money spent conducting network scanning and cross-checking license entitlements, it will continue to fail to address those disparate networks that maintain no logical connection to the MCEN-N or MCEN-S. Fundamental changes in governance and policy may be required to enable lower echelon commands to take responsibility for their license entitlements and SLM methodology, as was the case for several respondents within this study. Auditing these commands is likely necessary if the USMC moves forward with this policy. Such as the case with the Field Supply and Maintenance Analysis Office (FSMAO) auditing for supply inventory management, a separate organization may need to stand up to audit software inventory management.

For those logically visible networks from an enterprise perspective, the Marine Corps should continue to pursue a CMDB that retains the capability to communicate with its multitude of software discovery tools. At the time of this writing, ServiceNow appears to have a CMDB capable of doing so and accomplishes this concurrently with software packages offered by Tanium and Bel Manage, which are already in use (Service Now Inc., 2022). Determining the cost factor for introducing an entirely new suite of systems was outside the scope of this thesis; however, given the risks associated with further inaction in implementing a new system, it is worthwhile to explore ServiceNow as offering a potential solution.

Finally, underlying all of these challenges are factors that influence the need to require software licenses. Respondents in this study highlighted that specialized and project-based mission requirement, access to free and open-source software, availability

of “stand-alone” off network computer assets, and a lack of access to internet based tools contribute to increased risk of unauthorized software use. At the unit level, managers must be aware of the prevalence of these risk factors across all users in their command. The greater the awareness, the more proactive measures they can take to mitigate the risk of unauthorized software use. Future auditing teams, should they be implemented, can similarly identify these factors to determine how thorough their software inventory audit should be. When faced with a lack of options, users are expected to find a means to achieve mission accomplishment. This need to achieve is especially true in the high risk and high reward environment that Marines often face, both in garrison and deployed. Though some level of risk is necessary, it must be balanced with the cost of breaching a contractual agreement.

2. Solutions as They Relate to Organizational Change

Organizations must approach future changes to understand the risks and returns associated with them. In the context of this study, each proposed change and recommendation that the USMC can adopt reside in separate categories of the spectrum of change. This model, depicted in Figure 11, provides a visualization of the level of risk the USMC accepts when confronting change and the associated value that the change can provide.

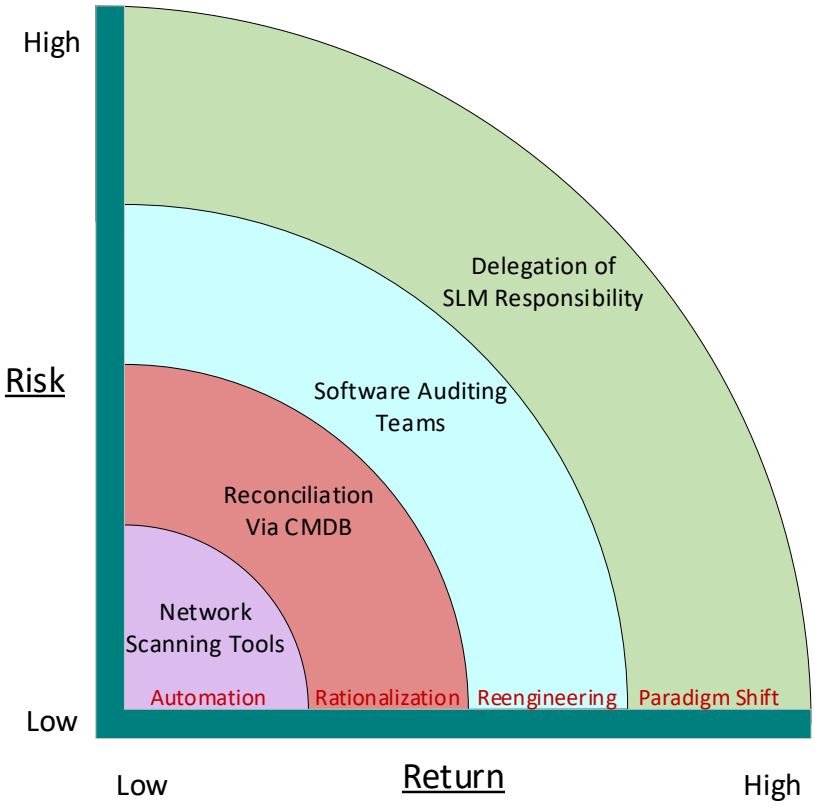

Figure 11. Spectrum of Change Visualization as it Relates to SLM Solutions. Adapted from Dixon (2016).

a. Automation

The main benefit provided by automation is speed. Current tasks already completed by hand are placed into a system that does the work with no required input from a user, and as a result, the time and money saved in the process can improve the overall efficiency of the process (Dixon, 2016). Software discovery tools such as Tanium, BMC Discovery, Big Fix, SCCM, and MDE are all considered elements of process improvement and change that fit within this category. They are considered low risk because, by and large, no significant organizational process changes are needed in order to implement them. These software programs are simply procured and placed within an already existing network that now does little more than provide discovery data upon request. As was discovered throughout this study, such applications are already in place in various organizations and, given their low risk of implementation, this is not

surprising. The essential takeaway is that while the risk of adding additional systems to assist with automation is low, so is the return in value that they provide. Discovery tools enable greater visibility for what exists on the network, but that visibility is all they truly provide to the organization. If anything more is needed to assist in license reconciliation, riskier levels of change are needed.

b. Rationalization

Through rationalization, organizations take existing processes or procedures and find the means to increase the value or efficiency that they already provide (Dixon, 2016). This change is separate from automation in that it goes beyond the requirement of simply speeding up a task and instead uses automation as a tool to seed more significant change. Integrating a CMDB capable of communicating directly with automated discovery tools falls into this category for several reasons. First, it is slightly more complex than automation in that it requires integration with other automation tools and provides a visible dashboard for license entitlement reconciliation. This information by itself does not provide any benefit to the organization without the right IT staff available to act on potential discrepancies. Second, the CMDB streamlines existing procedures for reconciliation by acting as a readily available and updated repository of existing software on the network and the status of their licenses. So long as it is designed to receive automatic updates, it simultaneously removes the time required to update manually and is a living source of reliable data that can be accessed at any time. Any processes that currently necessitate CMDB updates as a preliminary action are no longer bottlenecked by that requirement. Overall, the risks of introducing an entirely new CMDB software package, such as one provided by ServiceNow, are higher than those provided by available automation tools due to the cost and time involved in contracting and installing it. However, the potential returns in time, cost, and labor may make the investment worthwhile to explore.

c. Reengineering

In the context of business processes, reengineering requires a radical redesign of how a process functions overall, intending to eliminate repetitive tasks and dramatically

improve the quality of a particular service or product (Dixon, 2016). This stage of dramatic redesign of enterprise SLM maps relatively well with that of introducing an entirely new unit tasked with auditing digital inventory. The auditing team will exist for the purpose of improving the quality of SLM by direct enforcement of organizational policy in those networks where automated tools cannot reach. The risks associated with establishing this team are understandably higher in comparison to the previous two solutions, as it requires additional staffing, funding, and fielding in order for the program to function. However, there is little to no visibility for those disparate networks with no logical connection to the MCEN. Suppose an established auditing team can successfully enforce those networks into maintaining license entitlement compliance. In that case, the value it provides could exceed that of the risk associated with license fraud and non-compliance.

d. Paradigm Shift

The most radical form of change is the paradigm shift, where the general nature of a business process or mission set is altered to improve its strategic standing (Dixon, 2016). As an organization, the USMC has grown so that a single enterprise network is insufficient to conduct all of its necessary functions across the globe. Visibility across each of these networks is unfeasible when considering that many of them continue to exist in domains that do not even communicate with the MCEN. To this point, delegating the responsibilities of software license procurement, contracting, and management to lower commands may be necessary. Such a drastic shift in policy places significant risk on the organization as it will require time and funding to approve at the headquarters level, train those commands accepting responsibility, and routinely audit those commands to ensure they comply with SLM policies. If successful, the return of this paradigm shift will simplify SLM for the USMC, as only those networks directly operated and controlled by MCSC will require monitoring and compliance management at the enterprise level.

B. LIMITATIONS

Several factors inhibited the potential for more in-depth research regarding enterprise SLM and software discovery. Interviews were extremely valuable in gathering direct insight from those operating in the fields of network, systems, and IT asset management. However, this method did not provide the researcher with the ability to directly witness and test the systems tools discussed during these interviews. Such limitations placed heavy reliance on their perceived effectiveness from the respondents' viewpoints rather than through direct quantitative assessments of their actual capabilities. An inability to obtain real-time data from discovery tools was not a significant hindrance in understanding the fundamental issues at the core of enterprise SLM; however, many aspects regarding their performance were simply assumed as a result.

Funding, time, and travel restrictions placed considerable limitations on the number of interviews that were conducted and analyzed. Most interviews were conducted through virtual means, preventing the researcher from gaining additional research data regarding each organization's physical systems architecture and workspaces. Additionally, individual interviews took a significant amount of time to codify, analyze, and compare with one another to determine trends and best practices for SLM properly. This was the primary cause for the limited number of adequately interviewed respondents for research purposes.

Lastly, security concerns limited the amount of information that could be presented in an unclassified format over Microsoft Teams. This was especially true regarding questions describing past instances of unauthorized use of software and discussions that revolved around physical and logical specific network security measures. Respondents were also cautious of disclosing any information regarding particular software instances' specific business or mission functions. While this largely is unrelated to the overall purpose of understanding how to manage the licenses of those products, knowing their functions and restrictions may shed light on how to approach SLM for different product categories.

C. RECOMMENDATIONS AND FUTURE WORK

Based on the findings of this study, additional research is required to determine how government enterprises should approach SLM comprehensively. Reliance on third-party software use is at an all-time high, and there is no existing single standard for how vendors should supply their software packages to their customers. There is value in determining whether government organizations should establish a standardized contracting and license distribution practice, whether that be through subscription-based means or specified license entitlements. Otherwise, the rate of change within the private sector may continue to outpace any SLM methods that government entities adopt to the point that they never reach a sustainable level of cost-effectiveness.

Likewise, the applicability and usability of many of the discovery tools listed in this study remain uncertain. Future work should consider a quantitative assessment as to how effective these tools are in capturing discovery data from hardware and software within specific networks and whether that data can be placed into a transferable format that is legible in a single database repository or CMDB. This task could be accomplished by utilizing multiple testbed networks that exist on separate domains but still maintain some level of communications capability with one another to determine what minimum ports and protocols are required to accomplish this task. Tests for these software packages could also be completed utilizing a digital twin framework, where a virtual model of the MCEN and other disparate networks are created to represent the processes of their physical counterparts (Batty, 2018).

In addition to testing virtual tools, the USMC should consider testing a pilot program for a software inventory auditing team. This program must consider manning, cost, task organization, funding sources, and the overall value in contrast to the risk it prevents. It is much easier to quantify fraud, waste, and abuse of physical inventory. That level of ease does not translate well into the world of software, yet this does not justify ignoring the risk. Future theses can model what constitutes this pilot program prior to testing to provide the USMC with a further optimal baseline.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX. INTERVIEW QUESTIONNAIRE

1. Please state the organization of which you are a part of.

2. What is your position within your organization?

3. Is this position related to any network administration?

4. Select the response below that best describes your organization's structure.

   - Functional
   - Projectized
   - Matrix

5. If matrix, is it considered strong, medium, or weak?

6. Does the network structure or architecture efficiently map to the physical structure of the organization?

7. If any, please list some requirements for any network structure improvements.

8. Does your organization employ multiple networked domains? If so, how many currently exist that you either directly manage or work closely with?

9. Does your organization allow users to install third-party software onto networked computers without prior authorization?

   - Yes
   - No
   - Unsure

10. Are there any exceptions to this rule?

11. Has this regulation been relaxed during the COVID-19 pandemic?

12. Are there any ad hoc methods to monitor software installation during this process?

13. Have there been any noticeable trends among user experiences that indicate they more likely to utilize software that is not authorized on your network(s)?

14. Do members of the Information Technology (IT) team compromise between providing software solutions over better security standards?

15. What policies does your organization currently employ to prevent improper utilization of unauthorized software on your network(s)?

16. Of these policies, are any of them effective in their goals? Why or why not?

17. How often does your organization require users to participate in cyber security training?

18. Have there been any past instances where unauthorized software on the network created an incident that halted workplace productivity?

    - Yes
    - No
    - Unsure

19. Can you elaborate on any of these instances?

20. To what degree does your organization rely on third-party software to conduct many of its major business functions?

21. Can you give any examples of this third-party software?

22. If known, what were the functional, non-functional, and user requirements that led such software to be selected?

23. For any software mentioned previously, does your organization employ one license across the enterprise, or does it selectively choose which users to grant access to it?

24. For software that is not required for all users, how are those licenses managed? For example, are they distributed to the user's account or directly to their workstation?

25. What means are utilized to track and store software license information?

26. Overall, is software license management (SLM) considered a challenge within your organization? If so, can you explain how?

27. What tools does your organization utilize to discover and manage existing software on networked workstations?

28. Of these tools, are any of them considered to be effective?

29. Are any of these tools capable of collecting discovery data and storing it into a single external database?

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Adobe Inc. (n.d.). *Adobe Acrobat DC*: *Edit PDF files*. Retrieved March 28, 2022. https://www.adobe.com/acrobat/how-to/pdf-editor-pdf-files.html?mv2=Reader&DTProd=Reader&DTServLvl=SignedOut&ttsrccat=IPM*RDRDC-ALL-ACOM-202010*EN*DC0655*ExportRename*Var1

Alevizos, L., Ta, V. T., & Hashem Eiza, M. (2022). Augmenting zero trust architecture to endpoints using blockchain: A state-of-the-art review. *Security and Privacy*, *6*(1). https://doi.org/10.1002/spy2.191

Arora, A., Caulkins, J. P., & Telang, R. (2006). Research note—sell first, fix later: Impact of patching on software quality. *Management Science*, *52*(3), 465–471. https://doi.org/10.1287/mnsc.1050.0440

Bates, D. G. (2017, September 21). How to protect your company from an unlicensed-software crackdown. *The Business Journals.* https://www.bizjournals.com/bizjournals/how-to/technology/2017/09/how-to-protect-your-company-from-an.html

Batty, M. (2018). Digital twins. *Environment and Planning B: Urban Analytics and City Science, 45*(5), 817–820. https://doi.org/10.1177/2399808318796416

Binz, T., Breitenbucher, U., Kopp, O., & Leymann, F. (2013). Automated discovery and maintenance of enterprise topology graphs. *2013 IEEE 6th International Conference on Service-Oriented Computing and Applications*, 126–134. https://doi.org/10.1109/SOCA.2013.29

Defense Information Systems Agency. (n.d.) *Enterprise Mission Assurance Support Service (eMASS)* [Fact Sheet]. Retrieved March 2022, from https://www.disa.mil/~/media/Files/DISA/Fact-Sheets/eMASS.pdf.

Department of Defense. (2018). *Profile of the military community – 2018 demographics.* https://download.militaryonesource.mil/12038/MOS/Reports/2018-demographics-report.pdf

Dixon, S. (2016). *Business Process Reengineering*. *Slide Player*. https://slideplayer.com/slide/9543354/.

Dowd, M., McDonald, J., & Schuh, J. (2007). *The art of software security assessment: Identifying and preventing software vulnerabilities*. Addison-Wesley.

The Free Software Foundation, Inc. (2019, July 30). What is free software? *GNU Operating System*. https://www.gnu.org/philosophy/free-sw.html.en#translations

Gao, X. (2022). Competition between proprietary and open-source vendors with security concerns. *Technology Analysis & Strategic Management*, 1–13. https://doi.org/10.1080/09537325.2022.2045011

Ghaffarian, S. M., & Shahriari, H. R. (2018). Software vulnerability analysis and discovery using machine-learning and data-mining techniques: A survey. *ACM Computing Surveys, 50*(4), 1–36. https://doi.org/10.1145/3092566

Hall, K. (2011, February 14). Business Software Alliance fines company £24,000 for unlicensed Microsoft software. *TechTarget*. https://www.computerweekly.com/news/1280095145/Business-Software-Alliance-fines-company-24000-for-unlicensed-Microsoft-software

Haeussinger, F., & Kranz, J. (2013). Information security awareness: Its antecedents and mediating effects on security compliant behavior. *Thirty Fourth International Conference on Information Systems, Milan 2013.*

Hsieh, P.-H., & Lee, T.-K. (2012). 361 Does age matter? Students' perspectives of unauthorized software copying under legal and ethical considerations. *Asia Pacific Management Review, 17*(4), 1–19.

Hsieh, P.-H., & Yeh, K.-C. M. (2012). Cultural effects on perceptions of unauthorized software copying. *The Journal of Computer Information Systems*, *Fall 2012*(53), 42–46.

Hughlett, R. (2004). Firm incurs $77,644 fine for unlicensed software use. *Washington Business Journal, 23*(25), 1.

Kim, N. S. (2008). The software licensing dilemma. *Brigham Young University Law Review, 2008*(4), 1103–1164. https://digitalcommons.law.byu.edu/cgi/viewcontent.cgi?article=2422&context=lawreview

Kumar, A., Gupta, A., Sanagavarapu, L. M., & Reddy, Y. R. (2022). An approach to open-source software license management using blockchain-based smart-contracts. *15th Innovations in Software Engineering Conference*. https://doi.org/10.1145/3511430.3511448

Kumar, S., Biswas, B., Bhatia, M.S. and Dora, M. (2021), Antecedents for enhanced level of cyber-security in organisations, *Journal of Enterprise Information Management, 34*(6), 1597–1629. https://doi.org/10.1108/JEIM-06-2020-0240

Lahiri, A. (2011). Revisiting the incentive to tolerate illegal distribution of software products. *Decision Support Systems, 53*, 357–367. https://doi.org/10.1109/hicss.2011.363

Lindman J., Paajanen A., and Rossi M., (2010) Choosing an open source software license in commercial context: A managerial perspective, *2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications*, 237–244, https://doi.org/10.1109/SEAA.2010.26.

Liu, Z., Zhang, Z., Wang, Z., Peng, J., & Wu, S. (2021). Choosing an open source license based on software dependencies*. 2021 IEEE International Conference on Software Engineering and Artificial Intelligence (SEAI)*, 30–36. https://doi.org/10.1109/SEAI52285.2021.9477531

March, J. G., & Simon, H. A. (1958). *Organizations*. John Wiley & Sons.

Montgomery, J., & Mixon, E. (2020, November 9). What is a CMDB (configuration management database)*? SearchDataCenter*. https://www.techtarget.com/searchdatacenter/definition/configuration-management-database

Phillips, D. E. (2009, pp. xviii-57). The software license unveiled: How legislation by license controls software access. ProQuest.

Scuderi, C., King, A., & Toohey, S. (2021). [Issue Brief]. *Data consolidation OTA: ROI and cost avoidance projections*. PEO Digital.

ServiceNow, Inc. (2021). The ServiceNow configuration management database. https://www.servicenow.com/content/dam/servicenow-assets/public/en-us/doc-type/resource-center/data-sheet/ds-configuration-management.pdf.

ServiceNow, Inc. (2022). CMDB - Configuration Management Database. https://www.servicenow.com/uk/products/servicenow-platform/configuration-management-database.html

Synopsys, Inc. (2020, April 7). 5 types of software licenses you need to understand: *Software Integrity Blog*. https://www.synopsys.com/blogs/software-security/5-types-of-software-licenses-you-need-to-understand/

Toohey, S., Jackson, M., York, J., & King, A. (2020). *D-UNS Enterprise Discovery Solution* [Memorandum]. Marine Corps Systems Command.

Von Solms, S. (2005). Information security governance: Compliance management vs. operational management. *Elsevier, 24*, 443–447.

Watts, S., & Davis, S. (2018, February 5). Software License Management (SLM) explained [web log]. *BMC Blogs.* https://www.bmc.com/blogs/software-license-management/#.

Xue, B., Xu, F., & Warkentin, M. (2018). Critical role of ethical leadership on information security climate and employee ISP violation behavior. *WISP 2018 Proceedings*, *16*. https://aisel.aisnet.org/wisp2018/16

Acquisition Research Program
Department of Defense Management
Naval Postgraduate School
555 Dyer Road, Ingersoll Hall
Monterey, CA 93943

www.acquisitionresearch.net