INTEGRATING A MODEL-BASED SYSTEMS ENGINEERING APPLICATION WITH A GEOGRAPHIC INFORMATION SYSTEM FOR CYBER SUPPLY CHAIN RISK MANAGEMENT

Ian Peña

A Capstone Project Submitted to the
University of North Carolina Wilmington in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

Department of Computer Science
Department of Information Systems and Operations Management

University of North Carolina Wilmington

2024

Approved by

Advisory Committee

_____	_____
Lucas Layman	Jefferey Cummings

_____
Geoff Stoker

Accepted By

_____
Christopher Finelli

ABSTRACT

The system of interest for this project is a modern digital enterprise and its external operating environment. Currently, the design, deployment, and operation of enterprise cyber risk management strategies are not significantly reducing the level of success of motivated threat actors. New methods are needed for faster and more effective enterprise cyber risk management.

The National Institute of Standards and Technology (NIST) currently provides guidance on enterprise cyber risk management strategy design in Special Publication 800-39. This document calls for the creation of a senior cyber risk manager role responsible for creating an enterprise's cyber risk management strategy. While NIST describes a work process for designing a cyber risk management strategy, it does not make a specific recommendation for tools that will assist a senior cyber risk manager in their design work.

NIST SP800-39 was published in 2011 [1]. Subsequently in 2014, NIST published "On Enabling a Model-Based System Engineering Discipline" [2]. In 2020, NIST also cited a paper titled "Model-Based Cybersecurity Engineering for Connected and Automated Vehicles" as a best practice [3]. From this, it can be concluded that NIST is aware of model-based systems engineering and its usefulness when applied to cyber risk management strategy design. The gap area this project addresses is the application of model-based systems engineering for enterprise cyber risk management strategy design.

An enterprise cyber risk management strategy design needs to secure all five elements of its attack surface [4]. As a practical matter, this project's scope is limited to just one attack surface element, supply chain security. A proof-of-concept web application was developed for designing a supply chain cyber risk management strategy using a commercially available model-based system engineering application.

The selected model-based system engineering application does not currently support a geographic information system view, which is recommended for holistically visualizing a supply chain network. Research suggests that a holistic view minimizes decision-making risk based on incomplete or inaccurate information [5]. This gap was addressed by creating a middleware application which functionally generates the necessary view using supplier information entered into the model-based system engineering application.

Two use cases for the application were considered. The first use case is a student who is learning how to design a supply chain cyber risk management strategy. The second use-case is a senior cyber risk manager who is tasked with designing a supply chain cyber risk management strategy for the enterprise they are defending.

Successful validation of the contemplated proof-of-concept functionality has highlighted the benefit of this approach and enabled follow-up work to address the other remaining enterprise attack surface elements. A better tool for enabling enterprise cyber risk management strategy design will benefit all enterprises in their competition with motivated threat actors.

CONTENTS

INTRODUCTION

Motivation for Work

Constantly, there are reports of an enterprise being successfully attacked by a motivated threat actor. These attacks negatively impact enterprise operations and can come at a high financial cost. Enterprise cybersecurity continues to evolve as lessons are learned from cyber-attack incidents and as new technology becomes available.

Challenge

A modern digital enterprise is a large-scale, complex system of systems. This type of enterprise presents a large and porous attack surface that is increasingly being exploited with success. Senior cyber risk managers need to become more effective in managing enterprise cyber risk. This work focuses on developing better means for enterprise cyber risk management taking into consideration the three elements of work: people, process, and technology.

Identified Human-Factor Issue

Three human-factor issues were identified. First, cyber risk managers are human. Without tool assistance, they have practical limits in terms of how much information they can learn, retain, process, and act upon. This creates a human factor risk that can present as suboptimal decision-making when creating an enterprise cyber risk management strategy. A weak enterprise cyber risk management strategy is a vulnerability that can be more easily exploited by a motivated threat actor. With tool assistance and the use of reality abstraction modeling, cyber risk managers can process and act upon greater amounts of complex information in less time.

Second, senior cyber risk managers need to work both horizontally and vertically within an organization when developing an enterprise cyber risk management strategy. This presents challenges with regards to team dynamics, potentially limiting success. A few areas of team dynamics which must be considered for improvement include communication, shared knowledge, and group decision-making in support of a common objective. A poorly functioning management team produces vulnerabilities that can be exploited by a motivated threat actor.

Third, it is important to observe how future senior cyber risk managers are being trained. Currently, classes discussing cyber risk management strategy design are more focused on theory than practice. The students' limited training is then reinforced with experience gained while working in an enterprise operating environment. Cyber defenders learning on the job, in a live operating environment, can introduce vulnerabilities that may be exploited by a motivated threat actor. The creation of a training environment in which mistakes can be made without consequences will help facilitate enterprise cyber risk management strategy design.

Identified Work Process Issue

NIST provides guidance for managing enterprise cyber risks in NIST SP800-39. This document is titled Managing Information Security Risk: Organization, Mission, and Information System. It outlines a work process for framing and treating enterprise cyber risks with the end output being an enterprise cyber risk management strategy. The guidance in this document was issued in 2011 and is now incomplete due to changes in the enterprise operating environment. Drivers for these changes have evolved over time and include the use of new digital technology and services. Additionally, the scope of the enterprise's attack surface has now expanded to include the supply chain. A sub-optimal work process for designing an enterprise cyber risk management strategy enables vulnerabilities that can be exploited by a motivated threat actor.

Identified Technology Issue

NIST SP800-39 does not provide any recommendation on tools or applications for designing a cyber risk management strategy. Yet, the use of tools or applications for enabling work processes is an accepted and well-known practice today [6]. There are many tools which support business processes, though seemingly no tools which support senior cyber risk managers. NIST is aware of model-based systems engineering and its application for cybersecurity purposes [2]. NIST is also promoting model-based system engineering as a best practice for cyber risk management strategy design [3]. While difficult to quantify, it is worth considering how the lack of decision-making support tools or applications is currently limiting enterprise cyber risk management strategy design and deployment.

Proposed Solution

Research suggests that the best way for someone to learn about the structure, operation, and control of a modern digital enterprise is with a virtual learning environment and reality abstraction models [7]. While this type of environment can be created using a model-based system engineering application, more functionality is needed before it can successfully resolve the use cases identified below.

First, the model-based system engineering application needs to be configured and augmented for enterprise cyber risk management strategy design. Second, the model-based system engineering application needs to be capable of generating a network view of an enterprise and its external digital operating environment. Creation of this view seems best accomplished with a geographic information system. This would require the development of a middleware application to link the data in the model-based system engineering (MBSE) application with the view created in the geographic information system (GIS). Third, the model-based systems

engineering application needs to be developed to support the cyber risk management strategy design work process identified in NIST SP800-39.

While there are five elements in an enterprise's attack surface, the proof-of-concept system for this project focused on only one of them for practical reasons – The selected element was the enterprise supply chain. Focusing on just one attack surface element facilitated a shorter development lead time and resulted in fewer required technology products to build the system.

Solution Use-Cases

There were two primary use cases that this project aimed to address. The first use case was the improvement of the learning environment for students training in supply chain cyber risk management strategy design. The intent was to train students like they will work and then have them work like they were trained – at speed and with greater effectiveness. The second use case was to provide decision-making support for senior cyber risk managers in their effort to design a supply chain cyber risk management strategy. Both use cases were achieved, as there was fidelity between the tools and methods used in both the training and work environment.

Thesis

An augmented model-based system engineering application, as generally described, will enhance supply chain cyber risk management strategy design for both students and working senior cyber risk managers.

Relevancy

Capability improvements for enterprise cyber risk management will have wide-scale adoption if the proposed solution proves to be an improvement over traditional risk management approaches. The model-based system engineering approach to cyber risk management, as outlined in the thesis, has the potential to benefit every enterprise, as all enterprises must manage cyber risk within their supply chain.

LITERATURE REVIEW AND ANALYSIS

NIST Cybersecurity Documentation

Two NIST guidance documents have been identified as particularly relevant to the subject of cyber risk management theory. NIST SP-800-161r discusses comprehensive supply chain risk management frameworks. NIST SP 800-39 centers around enterprise cyber risk management. While both documents are helpful and thorough, SP 800-39 in particular is in need of an update. The document does not address the supply chain as a component of the enterprise attack surface. The cyber supply chain needs to be integrated into the overall enterprise cyber risk management strategy.

Model-Based Systems Engineering

Models are an efficient way to work with large-scale complex systems. They help to facilitate information exchange in support of the requirements, design, and analysis of a system [8]. Interactive models can be used to represent multiple different levels of abstraction or complexity in a system [7]. Model-Based Systems Engineering (MBSE) serves three primary purposes as it relates to this project, seen below.

1: An MBSE application allows for the parametric definition of a model. The GIS application provides a holistic view of the parametrically defined model, which is used to facilitate information exchange.

2: MBSE makes use of embedded functions to generate views for use case realization. This is especially useful for enterprise cybersecurity, as MBSE can help to reduce complexity in large scale systems of systems.

3: An MBSE application also handles the project lifecycle management process via a structured language like Lifecycle Modeling Language (LML).

Applied Graph Theory

Applied graph theory drives the GIS model, making use of abstraction techniques to reduce complexity. Rather than using detailed street or air navigation routes between suppliers on the map, "shortest path" straight lines are used. Waypoints on the map represent graph vertices and lines between the points represent graph edges. Abstraction minimizes visual overload, which compromises the effectiveness of the model [9].

Innoslate Support

Innoslate was the model-based systems engineering application of choice for this project. Official software documentation from Innoslate has been an essential reference throughout development, with the API documentation [10] and software installation [11] pages in particular being most helpful. Additionally, the Innoslate support team provided assistance via conference call in support of development efforts. As a result of guidance received from Innoslate, both the "cloud" and "enterprise" instances of the program were used together. The cloud instance was used for application design, while the enterprise instance was used as a production environment for application execution. The enterprise version of the software requires a paid software license, which the Innoslate support team provided free of charge for use in this project.

Google Earth KML and SimpleKML

Keyhole Markup Language (KML) is a markup language that is used to represent structured geographic data. It is a standard format used by many GIS applications, including Google Earth. KML is very similar to other markup languages like XML but is distinguished by its ability to include detailed metadata and styling for maps. KML is not without its limitations, however, such as its requirement for specific structure and syntax. This can be problematic when scaled up to large datasets [12]. As described below, the use of Python to generate KML

data resolves the issue of scalability, though it introduces new challenges with regards to translation between the languages. For example, some features of KML are difficult to represent in Python via the chosen library and some Python data structures must be transformed to ensure compatibility with KML standards.

"SimpleKML" is a Python library that greatly enhances the capabilities of KML by enabling the generation of map data through Python code [13]. The library is utilized in this project for the purpose of building the middleware application. The GIS model outlined in this paper makes use of several KML features: network links, folders, waypoints, and line strings. Each of these KML elements are a core component of the GIS model and need to be implemented in Python.

Azure Cloud Documentation

A Microsoft Azure cloud instance was created for use as a production environment. Innoslate enterprise and all other supporting software was hosted on this instance so that the output data could be accessed from the internet. During the setup of the Azure virtual environment, a written guide was used to assist in the setup process. The guide contains advice on resource group creation, virtual subnet setup, and IIS preparation [14].

METHODOLOGY

Phase I: Creation of an Integrated MBSE-GIS Application

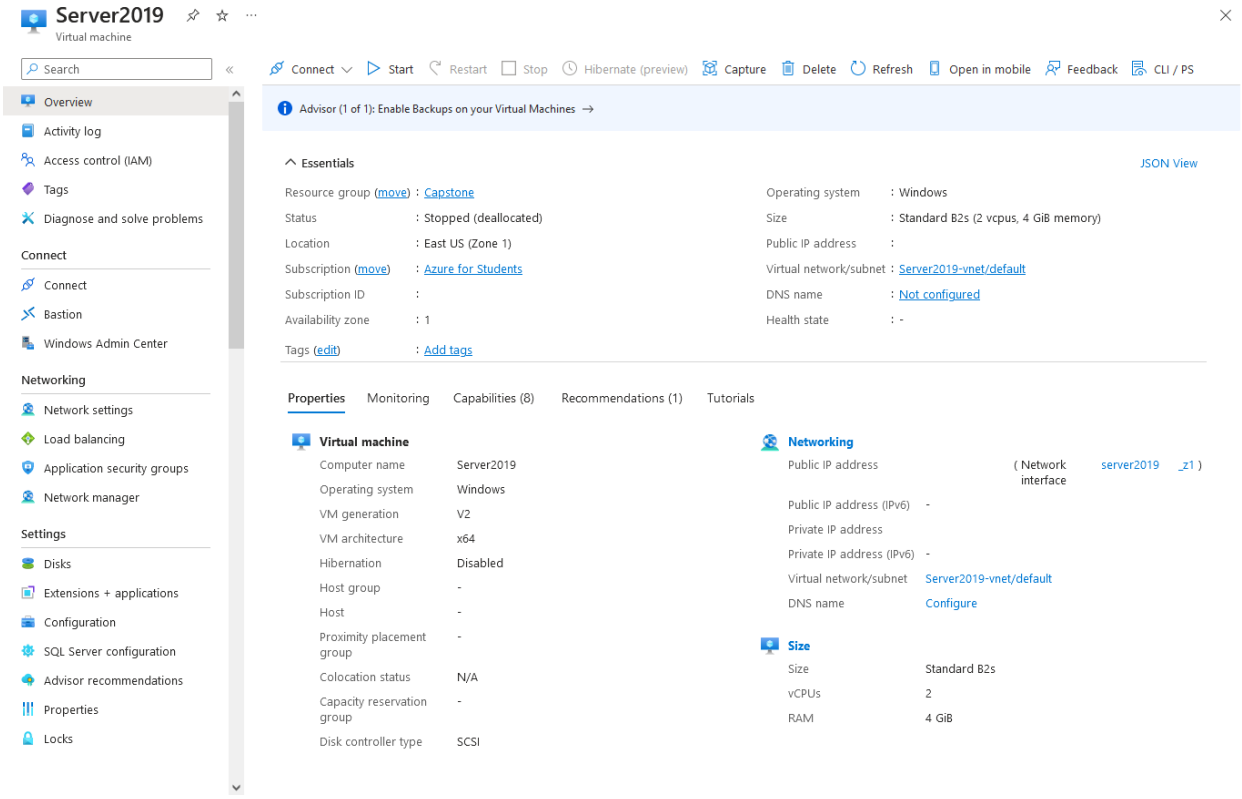
While Innoslate provides many useful views for enterprise cyber risk management, it lacks the capability to represent a holistic supply chain model. The supply chain model must instead be created in a GIS program. Google Earth was the GIS program of choice for use in this project, primarily because its license agreement allows for educational use free of charge. In order to bridge the gap between Innoslate and Google Earth, the proposed proof of concept middleware application is necessary. Each step in the creation of the middleware application is outlined below.

Part 1: Development Environment Setup

Azure Virtual Machine Deployment

In order to set up the Azure instance for use throughout this project, a series of configuration steps were performed. All Azure configurations can be performed from within the Azure web portal. The first task performed was creating a resource group for this project, which is a collection of compute resources and configuration values that are stored together for organizational purposes. Next, the virtual machine was created, and the operating system of choice installed onto it. That machine was then added to a subnet of a virtual network, with the ports for web and remote access opened. Setting up remote access via the Remote Desktop Protocol (RDP) required allowing the connection through the software firewall and then authenticating and connecting to the machine via the Windows Remote Desktop Connection application on my personal computer. Finally, important Windows settings within the virtual operating system were altered according to my needs, such as power usage, security controls,

and additional operating system features via the Windows Server Manager application. A screenshot of the Azure virtual machine page can be seen below.



Azure Virtual Machine Configuration Page

Innoslate Enterprise Installation

The Innoslate Enterprise program was installed to the virtual machine, along with all of its dependencies and requirements. After the program was installed, it had to be configured for first-time use. This process involved setting up API authorization, credential management, and filesystem settings.

Microsoft SQL Server Setup

Innoslate uses a relational database as a program backend, with support for either Postgres or Microsoft SQL Server (MSSQL). MSSQL was chosen for this project, as it offered better compatibility and integration with the rest of the technologies used throughout the project.

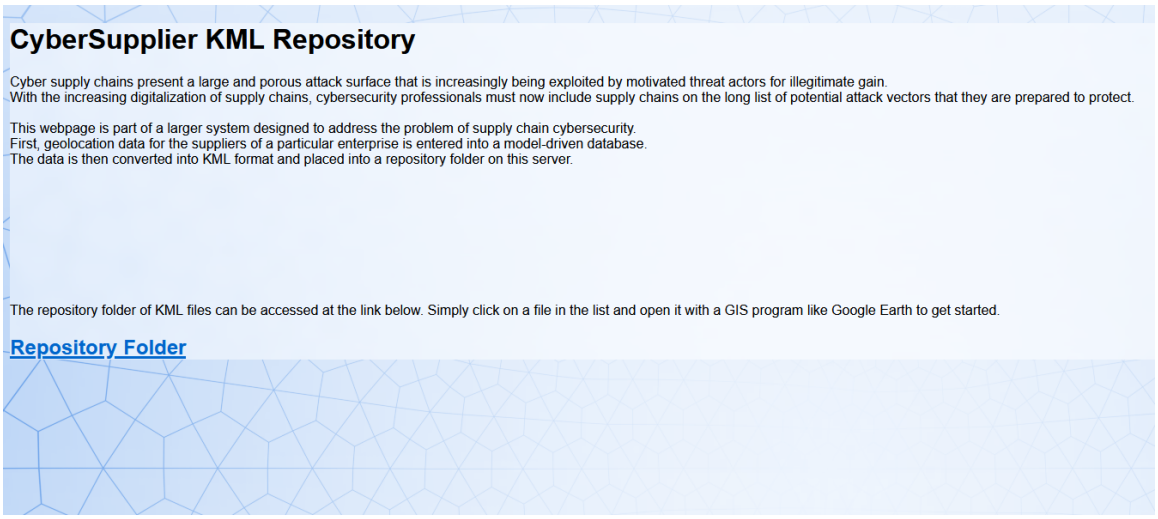
MSSQL was installed, with user accounts created and security policies altered to allow for connection to Innoslate Enterprise.

Development Environment Setup

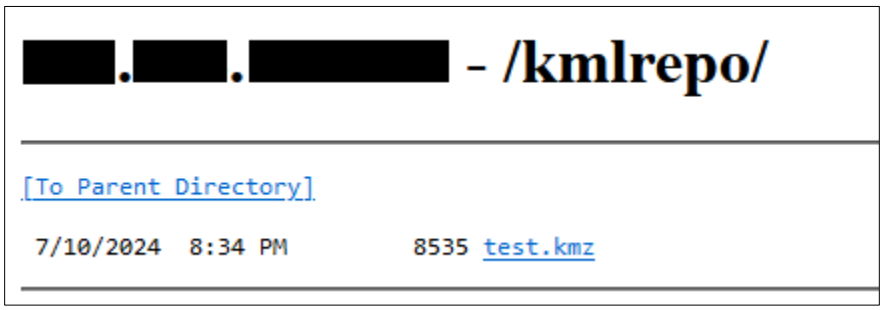
GitHub was used for this project to facilitate version control and code sharing. The code for the middleware application was hosted in a repository, which includes the executable code file, a configuration file, a README file, and an assets folder containing image files for map waypoints and legends. Although a "logs" folder is generated during program execution to store output, it is not included within the repository. PyCharm Professional was used as the IDE software due to its seamless integration with Git integration and simplified virtual environment setup.

Microsoft IIS Web Server Setup

On each execution, the middleware program would store the output files in a local folder on the virtual machine. To make the files accessible remotely, setting up a web server was necessary. Microsoft IIS was chosen for this purpose, as it is integrated into the Windows Server operating system and is easy to configure using the Azure configuration page. The middleware application was set to place the output files in a directory within the web server. Next, necessary settings such as MIME types and directory traversal were configured. Lastly, a basic HTML landing page was created. While both the “Link” and “Data” files are placed within the web server, only the “Link” file is accessible from the landing page. This is to prevent users from downloading the “Data” file, which would be incomplete on its own and lack the capability to retrieve updates from the middleware application. Screenshots of the web server can be seen in the figures below.



IIS Webserver Landing Page



IIS Webserver “Repository Folder” Containing a File

Part 2: Configuration of Innoslate for Intended Use

Innoslate Enterprise: Application Deployment

As mentioned previously, the enterprise version of Innoslate is required to run locally on the virtual machine, allowing for API access by the middleware application. In order to provide a framework for the imported supply chain data, Innoslate’s database schema needed to be modified. The creation of three new database classes was needed: Supplier, Interaction, and Boundary. Please reference Appendix E to see how each class corresponds to a feature of the

GIS model. After the classes were created, data importation was performed via the schema editor. Finally, labels were assigned to the data, enabling dashboard analytical views. A set of sample dashboard widgets has been created for this project and can be found in Appendix H.

Innoslate Cloud: Project Collaboration

Innoslate Cloud was utilized as a content sharing platform between members of the project. Content shared for collaboration purposes includes requirements documents, system diagrams, modeling, and research material. Innoslate Cloud required no additional configuration.

Part 3: Development of the Middleware Application

Development began by first deciding upon the program control flow. Due to prior experience with server applications in other projects, the decision was made to design the middleware application as a persistent command line window. The application remains running until the window is closed by the user. Next, file access functionality was built out, enabling the application to read configuration data and write KML output data. Once these steps were complete, the ability to perform API calls was put into place, with help from the Innoslate support team. Lastly, the application would perform the data conversion from API response data to KML model data. A screenshot of the middleware program in execution can be seen below.

CyberSupplier Middleware Application

Part 4: Creation of KML Files for GIS Usage

First, data was imported into the Innoslate database via the "Import Analyzer" feature. For the purposes of this project, a sample supply chain model has been created using data obtained from Corning Wilmington. Once the data was present, the configuration file needed to be altered according to the nature of this sample model. The data source and destination attributes were set, along with the necessary API queries. Once the application has been run, the output KML file can be retrieved from the web server and opened in the GIS program. The file on the web server is kept up to date with the database due to the "link" and "data" file distributed approach. The supply chain data is updated regularly for as long as the middleware application is running. For visual models describing this complex system in greater detail, please see the Application Architecture Diagrams found in Appendix D.

Part 5: Generation of Views for Use-Case Realization

A range of descriptive views are provided by the various components of the MBSE-GIS system. This includes both Innoslate and the GIS model. All views are listed below.

Innoslate Database Queries and Filters

On the "Database" page of the Innoslate interface, queries can be performed to filter suppliers based upon certain criteria. The selection of displayed columns can also be altered. If a particularly useful filtered query view has been created, it can be saved as a profile for later use. This representation of data is similar to most other traditional database software.

Innoslate Entity Diagrams

On the "Diagrams" page of Innoslate, system diagrams can be created containing entities and relationships. These values can be driven automatically by supplier records in the database. This was the method used to create the aforementioned diagrams in Appendix D.

Innoslate Dashboard Widgets

The "Dashboard" page is especially useful for referencing or presenting descriptive analytic data quickly and effectively. Like the diagrams, dashboard analytics are driven by database records automatically. Labels can even be applied to records to categorize them further. There are many different ways to visually represent analytics data, as seen in Appendix H.

GIS Supply Chain Model

This view is the central component of the entire MBSE-GIS system and is only made possible via of all of the previously mentioned supporting technologies. This model cannot be created through existing MBSE or database applications without tool assistance, which was what facilitated the need for the middleware application. The GIS model contains six different sub-views, each one aimed at assisting a different member of the supply chain management team.

The sub-views are selected within the places "Explorer" area of the GIS application as folders, which can be toggled on or off as needed. Each of the sub-views contains a unique map legend, which informs the corresponding team member of the color assignments for suppliers on the map. Screenshots of the GIS interface, supply chain model, and team member sub-views can be found within Appendix G.

Phase II: Application Verification/Validation Test

Test Objectives

Application requirements were recorded in Appendix C, to act as controlling records for performing the verification test. A successful verification test was achieved prior to the commencement of validation testing.

Validation testing was then performed in support of the two system use cases previously identified. In order to validate the product's use as a learning tool, three students were interviewed after interacting with the software. They were asked to reflect upon the ability of the software to improve the classroom experience for students learning to design a supply chain cyber risk management strategy. In order to validate its use as a communication tool, three industry experts were interviewed after interacting with the software. The experts were asked to reflect upon the ability of the software to help a senior cyber risk manager in designing a supply chain cyber risk management strategy. Both groups were asked to comment on the value of the application and provide constructive feedback for further application improvement.

In the realm of model-based system engineering, there exists the notion of a reference model – It has been proven to solve problems through experience [15]. A reference model solution gives the user a starting point with which to build from, rather than starting from scratch. Providing a student or senior cyber risk manager with an application that has an embedded reference model solution for designing a supply chain cyber risk management strategy

should reduce the time it takes to complete the design work and help facilitate a higher quality return.

Another anticipated benefit of performing the validation tests was that when considered collectively, they would provide insight into the feasibility of a unified software solution: A single application that addresses both the educational needs of students and the development needs of a senior cyber risk manager. This exploratory work is in alignment with research being conducted at MIT on management flight simulators [12].

Verification Test

Generally, the middleware application needs to be capable of generating a descriptive GIS view of a supply chain based upon supplier records in an Innoslate database. As seen in the list below, the functionality outlined in the application requirements document has been delivered upon. This suggests that the application meets design and build quality standards and is fit for intended use.

Delivered Functionality:

- ✓ Display a startup message containing environment variables.
- ✓ Run continuously in a console window until stopped.
- ✓ Output program activity to the console window in real time.
- ✓ Log program activity to a text file.
- ✓ Use a configuration file to determine environment variables.
- ✓ Fetch database records from Innoslate via API calls.
- ✓ Perform basic error checking.
- ✓ Produce a “Link” KML file to synchronize with “Data” file.
- ✓ Parse response data and produce a “Data” file.
- ✓ Create KML folders for each supply chain view.
- ✓ Create folders for Interaction Links and Security Boundaries.

- ✓ Generate Interaction Links and Security Boundaries using location and size attributes determined by database records.
- ✓ Generate KML waypoints for each supplier based upon their "Name" and "Location" attributes.
- ✓ Color KML waypoints depending upon selected folder view.
- ✓ Use specially labeled pins for the "Commodity" folder view.
- ✓ Produce a separate legend image asset for each of the 6 folder views.
- ✓ Populate waypoint descriptions with supplier properties.
- ✓ Track the total count of suppliers in the current view.

With all of the application requirements having been met, and the application verified as fit for use, validation testing was then performed.

Validation Test

Participant Information

All participants were given a brief background of the project, including the identified problem, use cases, and the developed software solution. Of the student participants, two had recently completed the CYBR 437 – "Supply Chain Security" course at UNCW. One student had no prior experience with supply chains or system modeling. The professional participants consisted of a wide range of experience. Professional 1 is a business analytics professor with extensive experience using databases in classroom environments. Professional 2 is an individual consultant on supply chain management. Professional 3 specializes in enterprise cybersecurity and military communications.

Demonstration and Follow-Up Questions

The specific questions chosen for inclusion in the validation test were selected to determine if the MBSE-GIS system was capable of demonstrating the points below. A handout sheet containing tasks to perform and questions to answer was given to participants. It can be

found in Appendix I. A technical limitation of this proof-of-concept system requires that participants enter supplier attributes into the system as integers. For this reason, a translation integer sheet needed to be handed to participants as well. The sheet can be found in the same appendix.

- Demonstrate the process of creating a dataset for supply chain team use through integration of multiple data sources.
- Demonstrate the value of the system as a holistic model of the supply chain, as it pertains to professionals on a supply chain management team.
- Demonstrate the effectiveness of using data driven KML scripts to generate a supply chain map automatically, as it pertains to students in a classroom environment.
- Demonstrate the significance of shared understanding and informed decision making, enabled by unique tailored views of the supply chain.

Results

The results of the validation test, as per each of the questions asked after the demonstration, can be found in the seven sections below:

1: Ease of Navigating the Innoslate Database

Students: Found the data creation/modification process easy and straightforward. Student 3 mentioned that the use of a code sheet for data entry may pose an issue to some, especially for the Commodity Code.

Professionals: Agreed that the process was easy but stressed the importance of consistent functionality and data integrity. Professional 2 suggested integrating with an ERP (Enterprise Resource Planning) solution for automated retrieval of scheduling information, which has already been considered as an area of future development. Professional 3 noted that in a real-world scenario, data would likely be entered into an existing supply chain management program, rather than entered directly into the Innoslate database.

2: Ease of Navigating the GIS Interface

Students: Found the GIS interface easy to use, especially appreciating that the legend images would change automatically depending on the selected view. Student 1 compared the folder views on the left side of the screen to using filters in other programs. Student 3 suggested making the pins larger for better visibility.

Professionals: Also found it easy to navigate. Professional 1 highlighted the need to consider the security risks of collecting supply chain data in a centralized location, especially when the model is so easy to access.

3: Usefulness of the Supply Chain Model

Students: Found the model helpful for gaining a better understanding of the supply chain. Student 2 stated that they are a visual learner, and as such, appreciated the ability to see suppliers of suppliers. Student 2 continues by stating they would have had a difficult time understanding the connection with international component suppliers if they were only looking at a database.

Professionals: Agreed that the model was a clear way of representing the supply chain, noting its potential for enhancing meeting presentations and strategic decision-making. Professional 1 liked that the analytical capabilities of the model were handled automatically by the database. Professional 2 emphasized the importance of visual aids for strategic supply chain planning and sourcing on a global scale. Professional 2 also observed that this model draws its strength from its ability to help people work differently and understand the "why" of their supply chain issues.

4: Effectiveness as a Training Tool for Students

Students: Believed the system would make an effective teaching tool, with the potential to make learning supply chain management easier and more engaging. Rather than needing to learn how to make supply chain maps manually, class members would be able to automate the

process. Student 2 points out that this would allow the class to focus more on the "big picture" significance of the supply chain makeup.

Professionals: Also believed that the system would improve the learning experience by providing realistic models of a sample enterprise to train on. Professional 3 states this model more closely resembles what prospective cyber risk managers will encounter in the real world. Professional 3 also points out that students would learn design principles like "lead time" much easier with a model to observe.

5: Effectiveness as a Communication Tool for Professionals

Students: Thought that the system would help ensure everyone is on the same page, reducing misunderstandings and facilitating collaborative work. Student 1 compared the functionality to an online shared document: Everyone can observe and work with the same set of information.

Professionals: Agreed that the system would likely enhance communication but noted that supply chain strategic decisions are not made daily. At most, they are made every quarter, and as a result, continuous real-time data synchronization should not be a development priority. Professional 2 emphasized the importance of human judgment and constructive observation, as facilitated by visual tools.

6: Functionality Requests

- Student 1: Enable use of built-in Google Earth search bar to find supplier information. Ensure that deleting a supplier waypoint also deletes relevant interaction links.
- Student 2: Place the total supplier count into an overlay image rather than in the explorer.
- Student 3: Use a dropdown selection within Innoslate rather than using a code sheet, if possible. Use different line colors to signify different types of interactions, such as the transfer of physical versus digital goods.

- Professional 1: Set up the database for displaying correlations between multiple data points. Develop capability to interface with other databases besides just Innoslate. Integrate other software products into the GIS model directly, using hyperlinks.

- Professional 2: Provide better GIS representation for matters of strategic importance. Enable a flat map mode in addition to the globe view. Add more views to the model for other potential members of the team, such as engineering, sales/marketing, etc. Generate interaction links automatically when a new supplier is created.

- Professional 3: Integrate the database with an ERP system for data automation. Integrate the GIS model with live network maps tools like "Visual Traceroute" or the "Live Cyber Threat Map". Overlaying one of these with the supply chain map would enable strategic decisions to be made based on geographic cyberattack data.

7: Additional Feedback

Students: Student 1 appreciated the minimal coding and supply chain knowledge required. Student 3 liked that runtime logs are kept by both Innoslate and the middleware application for safety and debug reasons.

Professionals: Professional 2 reiterated the need for consistent and up-to-date information while making decisions and acknowledges the system's ability to deliver upon it. Professional 3 noted that the more data points present in the model, the stronger its capability becomes. Cross referencing other security-related maps with the supply chain map can highlight discrepancies in the transfer of data.

CAPSTONE PROJECT DELIVERABLES

Business Requirements Document

Business requirements have been specified in a supporting document and should provide further context regarding the application of modeling techniques to supply chain risk management. They can be found in Appendix A.

Concept of Operations Document

A concept of operations document has been provided, which pertains to the original proposal for the GIS-MBSE augmented system. This document further discusses the relevance of the work and outlines the original approach towards the development of the middleware application. It can be found in Appendix B.

Application Requirements Specification

An application requirements specification was recorded before conducting the verification or validation tests. It is intended to improve the design and build quality of the application and ensure its fitness for intended use. The requirements specification can be found in Appendix C.

Application Architecture Diagrams

A series of entity diagrams have been included, which are dedicated to the composition of the middleware application and, more generally, the entire MBSE-GIS system. They serve to provide insight into the structure and internal functionality of the various systems included within the scope of this project. These diagrams can be observed in Appendix D.

Innoslate Functionality Map

Certain additions needed to be made to the Innoslate database schema in order to contain the imported supplier data. The views chosen for implementation in the GIS model drive the structure of the database schema. These database additions and the GIS functions they correspond to are provided in Appendix E.

Software Bill of Materials

A software bill of materials table has been maintained throughout the development of the MBSE-GIS system. It tracks all of the software components used in each part of the system and provides useful details about each, such as version number and license agreement. A software bill of materials is important for the sake of system documentation, troubleshooting, and security. See Appendix F.

GIS View Samples

The output of the middleware application is the GIS model view of the supply chain. Multiple screenshots of the GIS model, as well as the program interface have been included for reference. See Appendix G.

Innoslate Dashboard Widget Samples

An example dashboard layout has been created in the Innoslate instance used for this project. The widgets selected for the dashboard offer a visual representation of the data within the database. Some examples of these widgets are a recent activity feed, the total count of suppliers, and the buildout of supplier types within the enterprise. A screenshot of the dashboard is included in Appendix H.

Validation Report

For the handout given to participants during validation testing, see Appendix I.

CONCLUSIONS AND FUTURE WORK

Continuation of Work

Having evaluated the success of the project outcomes, it can be determined that there is a platform for continued research and development. Successful verification of application functionality has shown that the proposed system meets all stated requirements. Positive feedback during application validation testing has shown that there is promise of practical use in both educational and professional environments.

Technical Debt Management

If development were to continue, technical debt would be a serious consideration. Due to scope and time constraints, there was just one development cycle for the middleware application throughout this project. Future development would follow an iterative model, allowing for a reduction in code risk and complexity, as well as the opportunity for quicker user feedback.

There are known issues in the middleware application code, mostly due to software limitations that could not be fully addressed within the time scope of the project. There are also significant design changes that could be made to improve the quality of the software, such as improved error checking capabilities and the addition of more configuration options. More broadly, there would need to be a pivot from the monolithic nature of the codebase towards a modular design. While the vast majority of the project code is in a reusable state, the issues stated above warrant redesigning certain components before any further development occurs.

Improvements and Feedback

According to feedback from the participants of the validation test, there are many additional features which could provide value to the model if added in future revisions of the application. The model could be made easier to use with the utilization of existing GIS program

features such as the toolbar or the universal search bar. The model could facilitate understanding between even larger groups with the inclusion of additional team member views. The model could also provide more in-depth insights and analytics by integrating with other map systems to cross-reference data. Student and professional feedback obtained this early in the proof-of-concept stage will greatly influence the direction of future development efforts. Long-term application testing, such as usage in a classroom environment for the duration of a semester, would result in even more valuable data.

Next Steps

1: Full Coverage of the Enterprise Attack Surface

As stated previously, the scope of this project only addresses one of the five identified elements of the Enterprise Attack Surface, the supply chain. Integrating an MBSE application like Innoslate with additional programs has the potential to improve decision making capabilities for those respective elements. Information technology asset management software has been identified as an integration to address the "Digital Assets" element. Human resources management or payroll software could address the "People" element. This idea is explored at length in the last diagram in the list of Appendix D.

2: Creation of the Cyber Risk Management Workstation

By combining the model-driven database, the analytics dashboard, and the collection of model views, a cyber risk management workstation could be built. This workstation would be utilized by a cyber risk manager and would increase the efficiency and quality of decision making at the organizational scale. If fully developed, this workstation could potentially be sold as a standalone product.

Lessons Learned

Throughout the course of this project, I learned several important lessons regarding project management, software development, and academic research. One major area of growth was familiarizing myself with the tools and terminology used in the fields of systems engineering and enterprise cybersecurity. Given the scope and complexity of the project, effective time management proved to be a challenge. I have learned that when dealing with large bodies of technical work, sectioning off work into smaller tasks and working on multiple tasks in parallel can help to reduce time requirements. Another important lesson was the importance of confirming project expectations before initiating work to ensure that project objectives are aligned and understood between project stakeholders. Lastly, this project has taught me how to effectively communicate the business value of a product to potential users, ensuring that they understand the capabilities and use cases of the software solution.

REFERENCES

1: Joint Task Force Transformation Initiative. “NIST SP 800-39, Managing Information Security Risk: Organization, Mission, and Information System View.” *CSRC*, NIST, Mar. 2011, csrc.nist.rip/publications/detail/sp/800-39/final.

2: Denno, Peter, et al. “On Enabling a Model-Based Systems Engineering Discipline.” NIST, 28 May 2008, www.nist.gov/publications/enabling-model-based-systems-engineering-discipline. Accessed 25 May 2024.

3: Robles-Ramirez, David, and Theo Tryfonas. “Model-Based Cybersecurity Engineering for Connected and Automated Vehicles.” NIST, 23 Mar. 2020, www.nist.gov/system/files/documents/2020/05/04/MBE-2020_presentation_16_tryfonas.pdf.

4: Greer, Jeff, et al. *A Descriptive Enterprise System Model (DESM) Optimized for Cybersecurity Student and Practitioner Use*.

5: Sherwood, John, et al. Enterprise Security Architecture - A Business-Driven Approach. Routledge, 2021.

6: “What Is Business Process Management?” *IBM*, 17 Sept. 2021, www.ibm.com/topics/business-process-management.

7: Viste, Magnhild, and Hanne-Lovise Skartveit. (2004) “Visualization of Complex Systems - The Two-Shower Model.” *PsychNology Journal*, Department of Information Science and Media Studies, University of Bergen, 22 Feb. 2004, psychnology.org/File/PSYCHNOLOGY_JOURNAL_2_2_VISTE.pdf.

8: “The Ultimate Guide to Model-Based Systems Engineering (MBSE).” SPEC Innovations Blog, SPEC, 20 Mar. 2024, specinnovations.com/blog/guide-to-model-based-systems-engineering.

9: Lanum, Corey. *Visualizing Graph Data*. Manning Publications, 2017.

10: “Innoslate’s REST APIs.” *Innoslate Help Center*, Dec. 2023, help.specinnovations.com/innoslate-apis.

11: “Enterprise Installation.” *Innoslate Help Center*, Feb. 2024, help.specinnovations.com/enterprise-4.9-installation.

12: Wernecke, Josie. The KML Handbook: Geographic Visualization for the Web. Addison-Wesley, 2009.

13: Lancaster, Kyle, and Patrick Eisoldt. “SimpleKML Documentation.” *Overview*, 2021, simplekml.readthedocs.io/en/latest/index.html.

14: Ju-Shim. “Quickstart - Create a Windows VM in the Azure Portal.” *Microsoft Learn*, 23 Apr. 2024, learn.microsoft.com/en-us/azure/virtual-machines/windows/quick-create-portal.

15: Borky, John, and Thomas Bradley. *Effective Model-Based System Engineering*. Springer International Publishing, 2019.

16: Sterman, John. “Interactive Web-Based Simulations for Strategy and Sustainability: The MIT Sloan LearningEdge Management Flight Simulators, Part I.” *Climate Interactive*, System Dynamics Review, 23 July 2014, img.climateinteractive.org/2015/01/Learning-Edge-MFS-Part-I.pdf.

APPENDICIES

Appendix A
Business Requirements Document

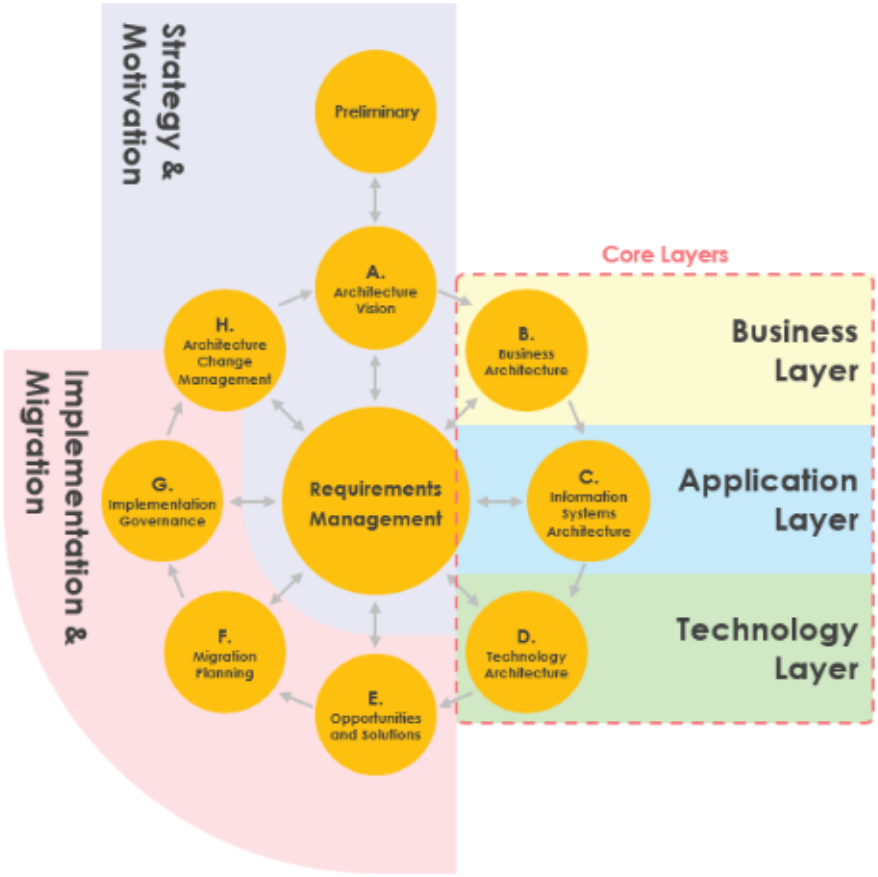
---

1 Document Purpose

This business requirements document will be used by Ian Peña to scope his capstone project deliverables. – Jeff Greer, 03/14/2024

2 Design Responsibilities

Reference the Open Groups Architectural Development Method (ADM) TOGAF.



Rationale: Jeff Greer, based on his research and prior work experience, will be responsible for developing business layer requirements. Ian Peña, based on his computer science knowledge, will be responsible for developing the application and technology layer requirements.

3 Background Information

3.1 System of Interest Declaration

The system of interest will be a modern digital enterprise.

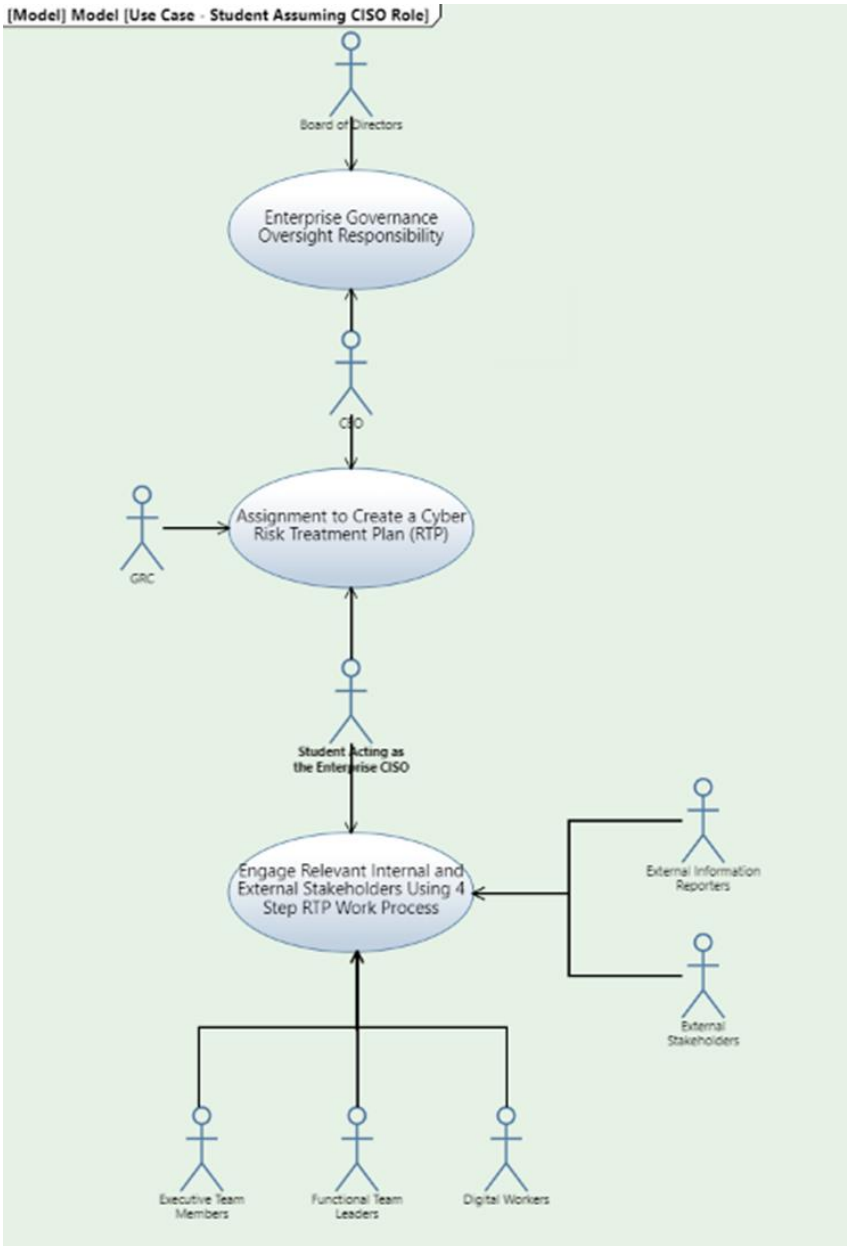
Rationale: A modern digital enterprise will be one of the following: 1) a for-profit business, 2) a non-profit organization, 3) a government entity, or 4) a DoD military branch of service. What they all have in common is a digital strategy deployed for enterprise mission achievement.

3.2 Observed Problem

Every day, there are news reports of successful enterprise cyber-attacks with negative impacts. The key question which will need to be answered is why? Afterall, enterprise cyber risk management theory is generally known and widely available for use. Is the theory underdeveloped or not properly applied?

Rationale: Currently, enterprise cyber risk management is learned experientially by students, post-graduation. This occurs because it is difficult to replicate the enterprise work environment in a classroom setting. Furthermore, working senior cyber risk managers do not have purpose-built tools supporting enterprise cyber risk strategy design and management.

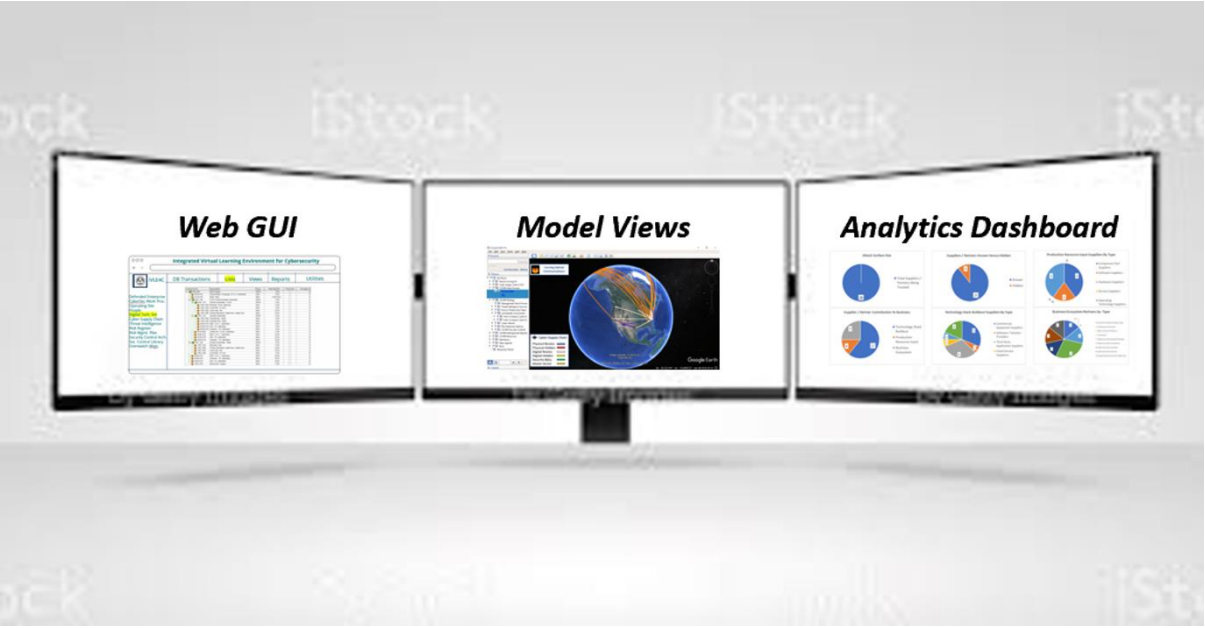
3.3 Use Case



Rationale: There is merit in seeking new means for improving senior cyber risk manager performance. It is important to consider both how students are trained to manage enterprise cyber risk and how working professionals perform while on the job. Targeted outcome: Train students like they will work, then once employed, enable them to work like they were trained at speed and with greater effectiveness when managing enterprise cyber risk. Fidelity between the training and work environment is needed for this outcome to be achieved.

3.4 Proposed Solution

A proof-of-concept dual-use (training and work) engineering workstation application will be developed and evaluated for effectiveness in assuring enterprise cybersecurity by design.



Rationale: The application will provide decision making support for practicing security by design. It will support guidance provided in NIST SP800-39 for Information Security, as well as guidance provided by INL for operating technology security.

Appendix B
Concept of Operations Document

---

1 Executive Summary

The concept of operations outlined in this document addresses the critical intersection of cyber supply chain security and model-based systems engineering. As enterprises navigate the complexities of modern digital ecosystems, the need for a comprehensive model-based approach becomes increasingly clear. This document outlines the importance of the proposed software solution, which achieves its mission via the integration of model-based systems engineering software and geographic information system software. Aimed at enhancing communication between teams across the enterprise, the software that I plan to develop will be a vital tool in the hands of a cybersecurity risk manager.

2 Introduction

The cyber supply chain is purpose built to fulfill enterprise needs for goods and services by delivering value to the customer. Every digital enterprise has a cyber supply chain, which introduces yet another attack vector for attackers to exploit, and therefore, increased risk.

In order to support the continuous operation of the enterprise, cyber risk managers must be able to visualize all aspects of the cyber supply chain, ensure compliance with all requirements, and mitigate risks through informed decision-making. By embracing security by design and leveraging advanced modeling techniques, the proposed solution seeks to optimize system performance, cost, and schedule while fostering a shared understanding across disciplines at the organizational level. This is achieved by providing insights into the operational scenarios, and the development methodologies of the organization.

2.1 Capability Need

The envisioned middleware application aims to bridge the gap between Model-Based Systems Engineering (MBSE) and Geographic Information Systems (GIS). The need for this software tool is twofold.

1. Facilitating knowledge sharing, which leads to higher quality decision making. Every member of a supply chain team needs to look at the same set of data, holistically, in order to have a shared understanding of the system. If multiple people on a cross-functional team are in a meeting, making important decisions based off incomplete or fragmented info, that creates a risk.

2. Employee turnover is another factor that creates a great deal of risk. Someone who is completely unfamiliar with the organization will need to learn the supply chain makeup from an enterprise cybersecurity perspective and do so as quickly as possible. The proposed software solution will meet these needs, aiding in the onboarding process of new employees.

By providing these capabilities, the middleware becomes essential for fostering a shared representation of the supply chain, optimizing decision-making processes, and ultimately producing more resilient digital ecosystems at the organizational level.

2.2 Current Situation

Innoslate is a very useful tool for cyber risk managers but does not have the necessary functionality to interface with a GIS system. If a risk manager wishes to use these two systems in conjunction, they must move data back and forth manually. In the proposed model, Innoslate allows for the generation of supplier data, while Google Earth provides the descriptive views. This proposed middleware application connects Innoslate and Google Earth so that a cyber risk manager can use them together with automation.

3 Operations and Support Descriptions

3.1 Missions (Primary / Secondary)

- **Primary mission:** Improve communication between members of cross functional teams, allowing for a shared understanding of cyber risk.
- **Secondary mission:** Serve as a learning tool with which aspiring cyber professionals can train on a realistic model of an enterprise.

3.2 Users and Other Stakeholders

The Stakeholders of this project are listed below.

- I, the developer of the software.
- The advisors on the board for this project.
- The users - cyber risk managers who would use this program.
- Future UNCW students who are interested in continuing related work.
- Innoslate, for providing the licensed software and offering support assistance.

3.3 Policies, Assumptions and Constraints

3.3.1 Policies

In support of NIST SP 800-39, this project will provide cyber risk managers with:

- The ability to develop better cyber risk management frameworks.
- A reduction in strategic and tactical risk across all three tiers of the organization
  - [See figure 2 on page 9 of the NIST document.]
- Improvement of communication between cross-functional teams.
- The ability to more clearly define critical business processes and how to defend them.
- Assurance that organizational resources are used and spent responsibly.
- Assurance that critical cyber risks are brought to the attention of decision-making authorities within the organization.

3.3.2 Assumptions

The success of this project assumes that:

- Innoslate Enterprise provides all of the necessary features for this project, and that the response data from API calls can be consistently captured and parsed.
- Google Earth Pro and the KML framework provide the flexibility and feature completeness that is needed to accurately represent an enterprise model geographically.
- The SimpleKML library performs its tasks completely and consistently, and remains a secure, supported, and up-to-date software library.

3.3.3 Constraints

Some of the constraints and restrictions of the project environment are:

- Innoslate Enterprise is required, as the cloud version does not support REST API calls. More information about API support and functionality can be found at this link: https://help.specinnovations.com/innoslate-apis
- An SQL server must be running at all times on the same machine as Innoslate Enterprise. This requires that certain security settings be altered, like port configurations, firewall settings, etc.
- Without linking to an ERP system [considered in future work] the user needs to manually import data into the Innoslate system from their own database.
- Only one dataset will be used for testing, in the interest of time. Compatibility would need to be verified with other datasets in future work.

3.5 Potential Impacts

The proposed middleware application has the potential to drastically improve communication across the organization, allowing for enhanced decision making both in terms of quality and speed. This can include purchasing decisions, asset management considerations, and the selection of suppliers. Additionally, the cross-functional teams who are responsible for supply chain management need to be able to visualize and understand the entire system of systems at the enterprise level in order to act effectively. Each member of the team comes from a different background, and as such, has different priorities when reaching important decisions. This application aims to unify each member under a shared understanding of the system.

4 Functional Capabilities

Utilizing Bower's Box method for determining system requirements, the proposed middleware application will:

- Receive data from the Innoslate database via API “GET” calls.
- Read the JSON response into Python and convert it into KML format.
- Save those files in a dedicated location and host them on a web server.
- When requested by a Google Earth client, distribute the KML files.
- Allow for the KML files on the client’s PC to sync changes with the remote data.
- All data that enters and leaves the system is formatted correctly.
- If one or more components of the network are down, the system fails safely.
- Indicators are provided for verification that the system is functioning as intended.
    - For example, a "last updated" field displayed upon each system interaction.

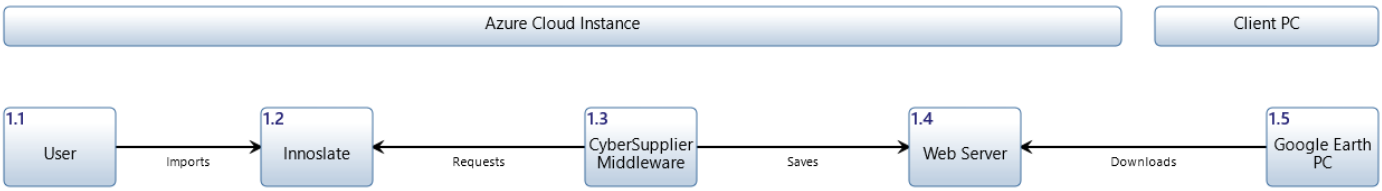
Appendix C

Application Requirements Specification

1: When executed, shall display a startup message containing environment variables.
2: Must run continuously in a console window until stopped.
3: Must output program activity to the console window in real time.
4: Must also log program activity on each run to a new text file, including timestamps.
5: Will parse a configuration file to determine environment variables, rather than expect hardcoded values.
6: Will fetch database records from a local installation of Innoslate Enterprise via API calls.
7: Must return meaningful error messages if API authorization fails, or if response data is malformed.
8: Must produce a “Link” KML file.
    a. Link file shall contain a link pointing to the “Data” file [see point 9] in a remote location and sync data with that file continuously.
    b. Must contain static branding assets, such as a logo image.
9: Must parse response data and produce a “Data” KML file.
    a. Data file shall create a new KML folder for each of the members of the supply chain team, each to be toggled on or off.
    b. Must also create folders for Interaction Links and Security Boundaries.
    c. Will create Interaction Links and Security Boundaries using location and size attributes determined by database records.
    d. Will create KML waypoints for each supplier based upon their “Name” and “Location” attributes.
    e. Will color each KML waypoint differently for each of the folder views, determined by corresponding attributes in the database.
    f. Will create specially labeled pins for the “Commodity” folder view.
    g. Must produce a separate legend image asset per each of the 6 folders, describing the significance of each pin color in each folder view.
    h. Shall place the values of a supplier’s attributes into that supplier’s waypoint description on the map.
    i. Shall track the total count of suppliers in the current view.

Appendix D
Application Architecture Diagrams

1: Overhead view of the entire system that is within the scope of this project.



User:
The user of the system is a senior cyber risk manager or cyber architect.

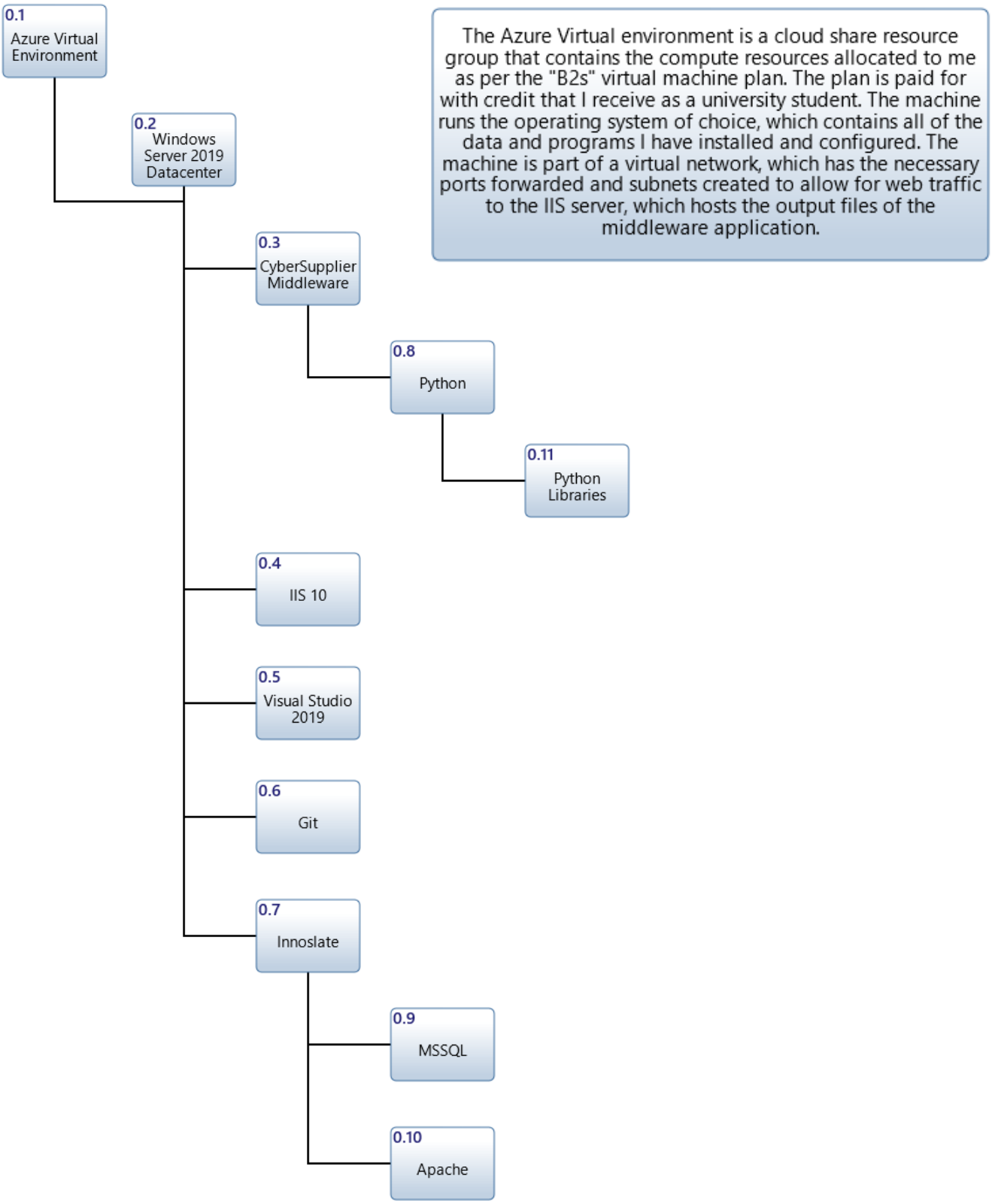
Innoslate:
Runs as a background service and can be accessed on localhost. Contains data input from the user. Accepts API requests.

CyberSupplier Middleware:
Takes in data from Innoslate via API requests. Converts response data into KML. Saves it to a folder in the Web Server.

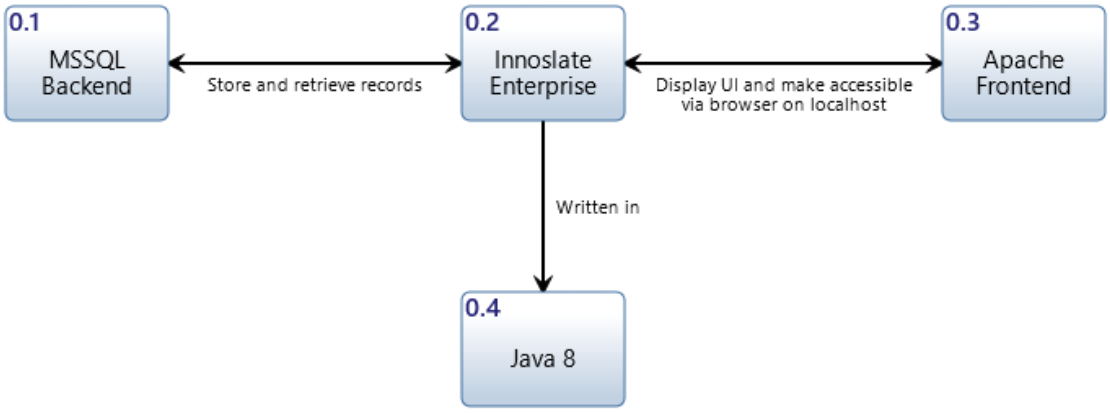
Web Server:
Contains the HTML to display the home page and two subfolders. One folder contains the output KML file with the network link. The other folder contains the KML file with the data from Innoslate.

Google Earth PC:
This is the device that will run Google Earth, it can be thought of as the "client" in this model. The user navigates to the public address of the Web Server using this PC and downloads the KML file containing the network link.

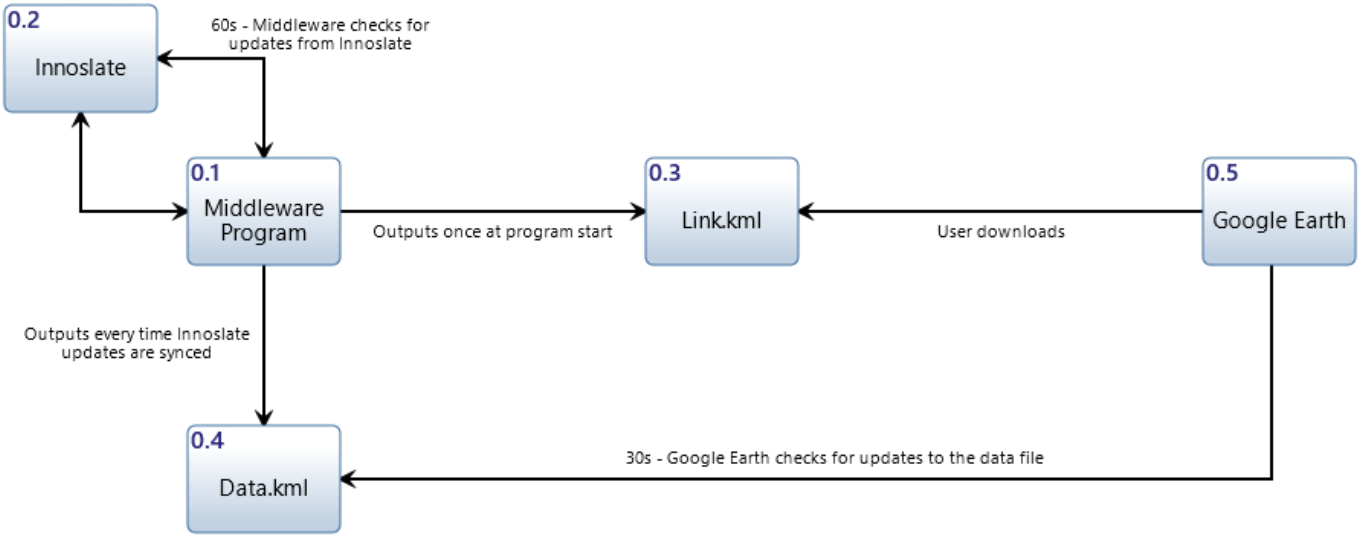
2: Hierarchal representation of the technology stack on the Azure cloud instance. Includes all software needed to run Innoslate, the middleware, and the web server.



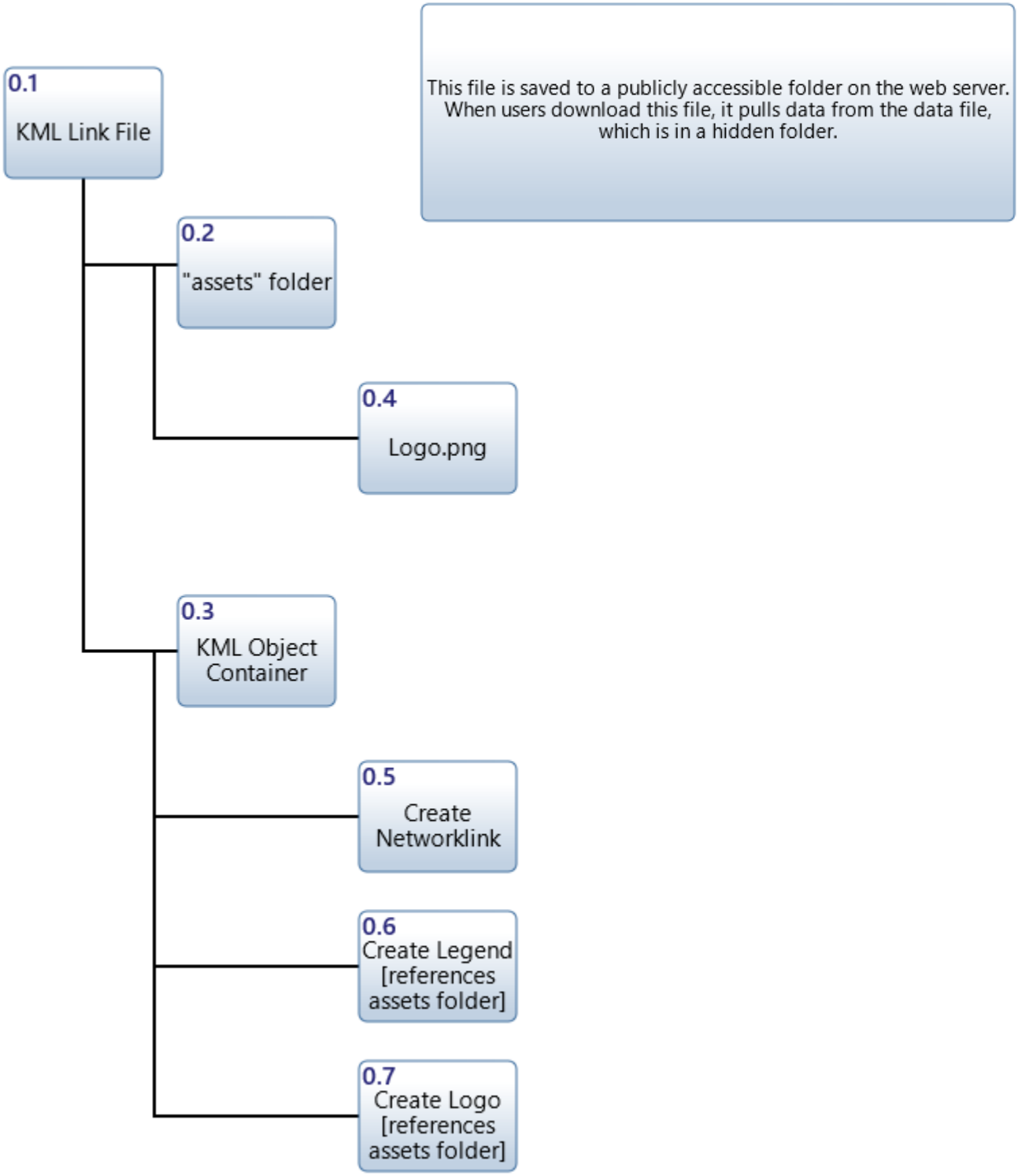
3: Decomposed view of Innoslate's requirements and capabilities.



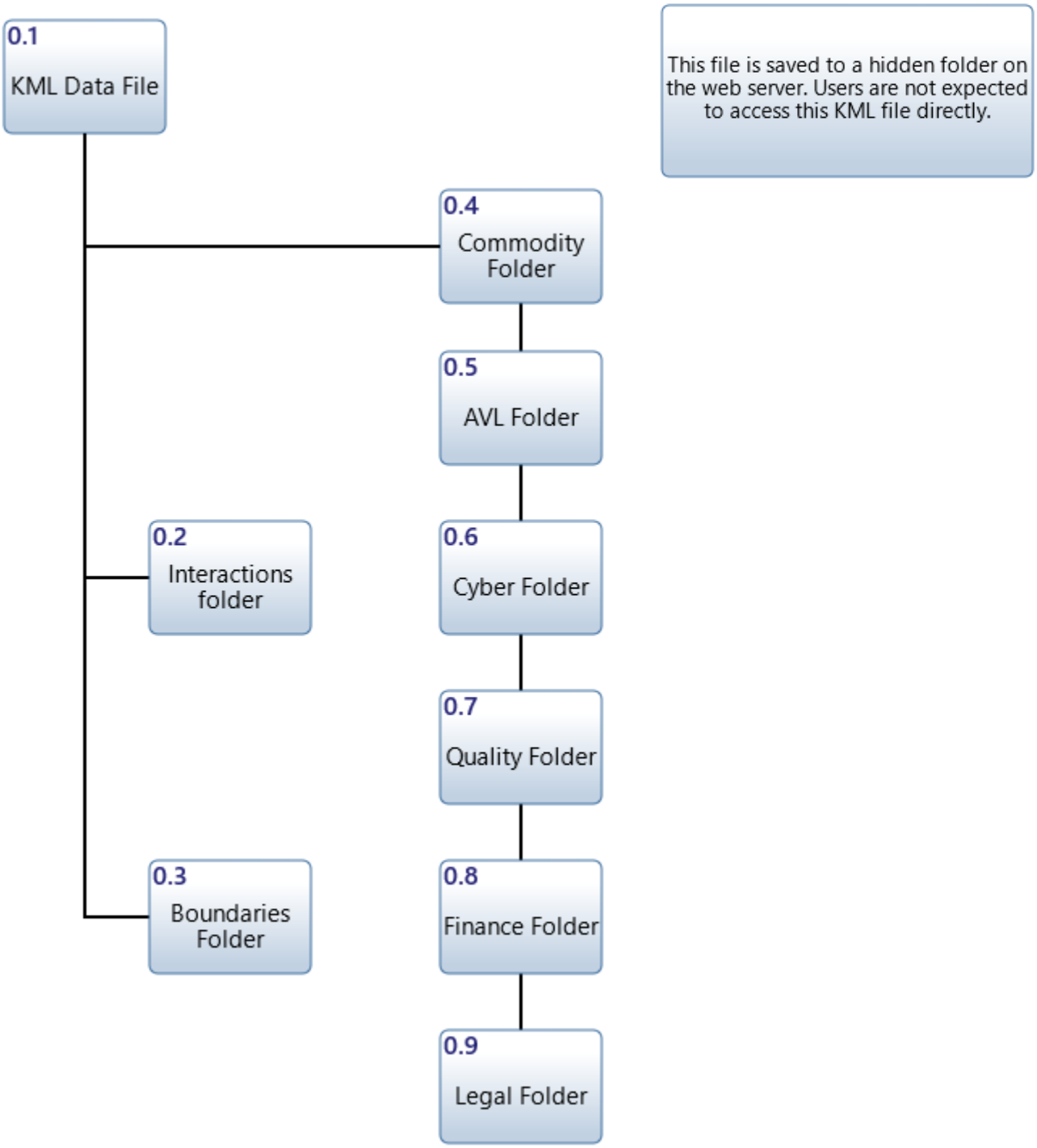
4: Decomposed view of the CyberSupplier middleware application.
This control flow shows the movement of data throughout the system.



5: Hierarchal representation of the “Link” output KML file showing all of its contents.

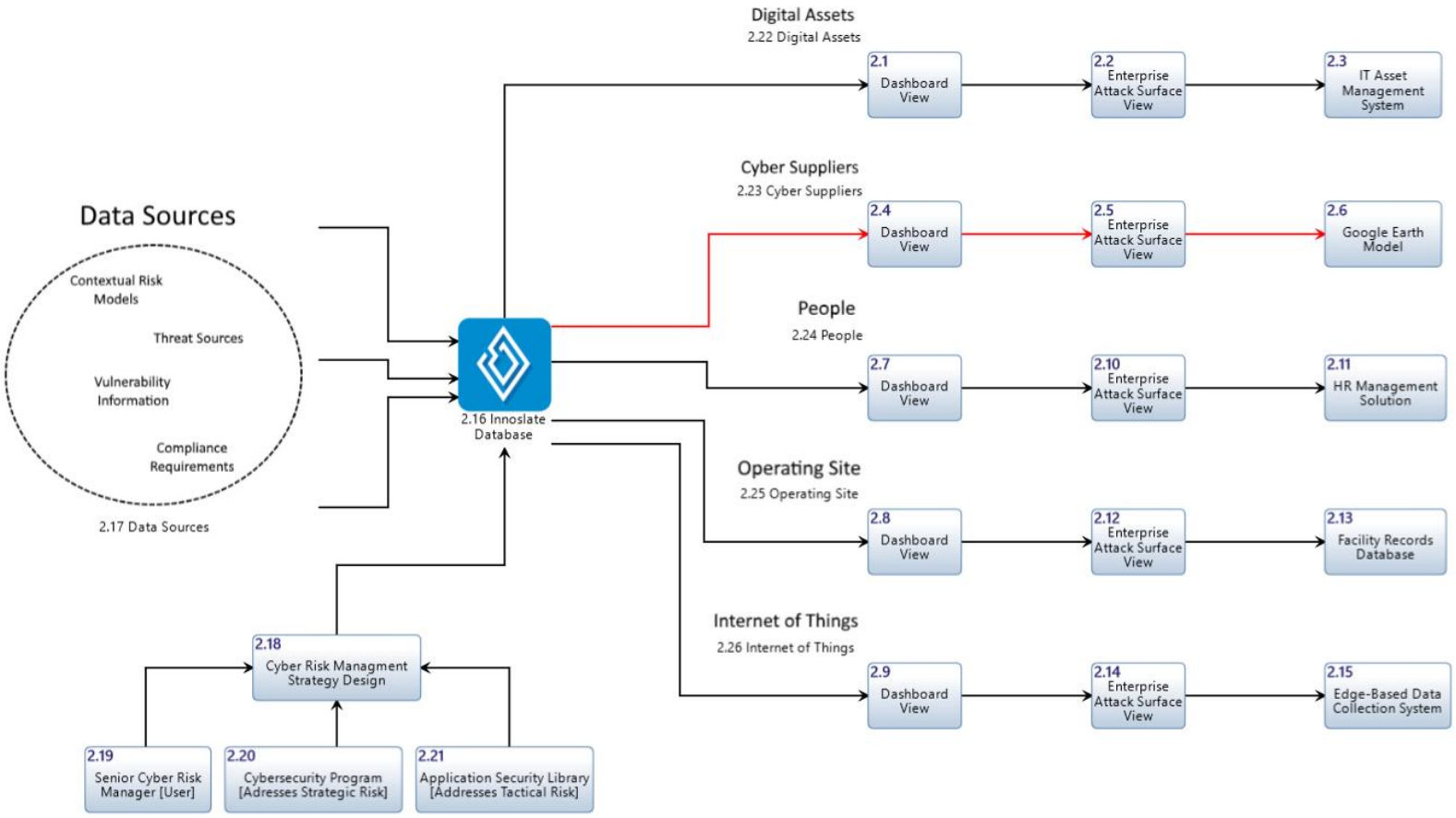


6: Hierarchal representation of the “Data” output KML file showing all of its contents.



7: Overhead view of the proposed future system for full coverage of the enterprise attack surface, allowing for comprehensive risk management strategy.

The red line represents the scope of this project.



Appendix E

Innoslate Functionality Map

Supplier Class

Innoslate Attribute	Google Earth Representation
Name	Title of KML waypoint object
Location	Latitude and Longitude position of KML waypoint
Commodity Code	Selection of title image to use for Commodity waypoints
AVL Status	Color of waypoints within AVL Status folder
Cybersecurity Trust	Color of waypoints within Cybersecurity Trust folder
Quality Assurance Plan	Color of waypoints within Quality Assurance Plan folder
Financial Credit Rating	Color of waypoints within Financial Credit Rating folder
Legal Agreement Type	Color of waypoints within Legal Agreement Type folder

Interaction Class

Innoslate Attribute	Google Earth Representation
Name	Title of KML line string object
From Location	Starting point of KML line string [Lat + Long]
To Location	Ending point of KML line string [Lat + Lon]

Boundary Class

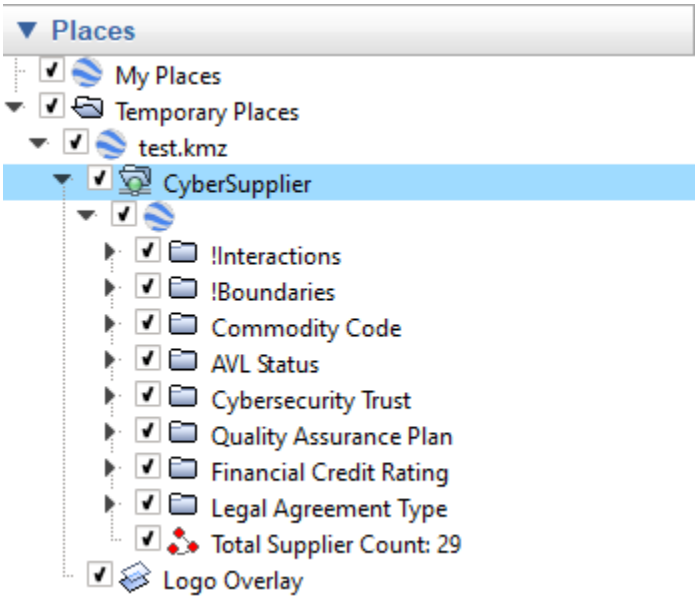
Innoslate Attribute	Google Earth Representation
Name	Title of KML circle object
Location	Center point of KML circle
Radius	Size of KML circle

Appendix F
Software Bill of Materials

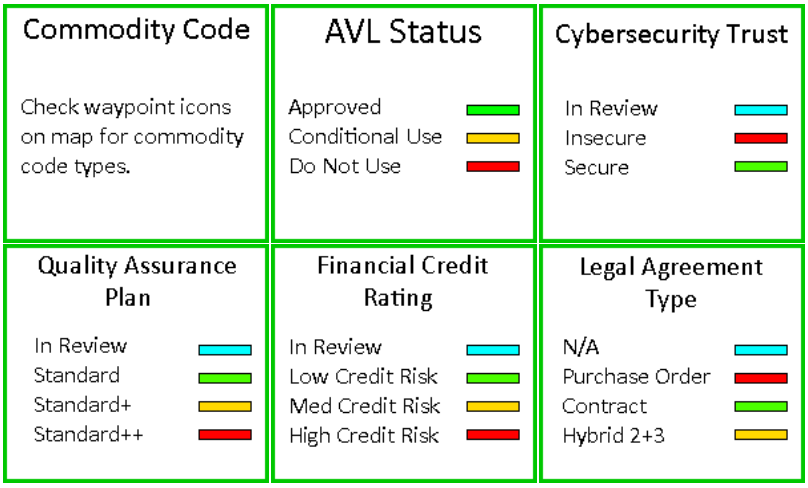
ID	ITEM NAME	VERSION	DESCRIPTION	LICENSE	MAKE	USE	BUY
1	Custom Middleware App	1	Performs API requests to Innoslate and retrieves entity data. Uses Python to translate the data into KML scripts. Saves the scripts in folders on a web server where they can be fetched by a client running Google Earth.	No license	✓	✗	✗
2	Microsoft Azure Cloud	N/A	Cloud compute platform that allows for remote hosting of the operating system, virtual network, and all of the software components in this list.	Educational	✗	✗	✓
3	Windows Server 2019 Datacenter	NT 10	Enterprise-ready operating system that runs on the Azure cloud share.	Educational	✗	✓	✗
4	Internet Information Services	10	Simple webserver that hosts the KML files generated by the middleware app.	Free with conditions	✗	✓	✗
5	Innoslate Enterprise	4.9	Cloud based data-driven approach to MBSE. Supplier records are imported into the program and then passed on to the middleware app. It is bundled with a version of Apache, which is used as the program's frontend.	Educational	✗	✗	✓
6	Microsoft SQL Server 2019	19.3	Database backend used by Innoslate to store entity records and other information.	Educational	✗	✓	✗
7	Java	8_201	Programming language that Innoslate is written in. Installed as a dependency.	Free with conditions	✗	✓	✗
8	Google Earth Pro	7.3.6.9750	Create and view maps with advanced modeling tools. Used to visualize supplier records geographically.	Free with conditions	✗	✓	✗
9	PyCharm Professional	2023.3/ 2024.1	Python IDE that simplifies development. Used to create the middleware app.	Educational	✗	✓	✗
10	Visual Studio 2019 Community	16.11.34	Product management software and multilanguage IDE used for automation of code deployment and inspection of output KML files.	Free with conditions	✗	✓	✗
11	Git	2.44.0	Used for version control, code backup, and deployment to the Azure cloud share.	GNU Lesser General	✗	✓	✗
12	Python	3.11	Programming language used by middleware. Supported by PyCharm. Installed on the Azure cloud share so that the program can run.	Open Source	✗	✓	✗
13	Simple KML	1.3.6	Python library that automates the generation of KML scripts. Translates Python to KML.	GNU Lesser General	✗	✓	✗
14	Other Python Libraries	N/A	configparser, datetime, logging, os, requests, time, urllib, polycircles	Assorted Licenses	✗	✓	✗

Appendix G
GIS View Samples

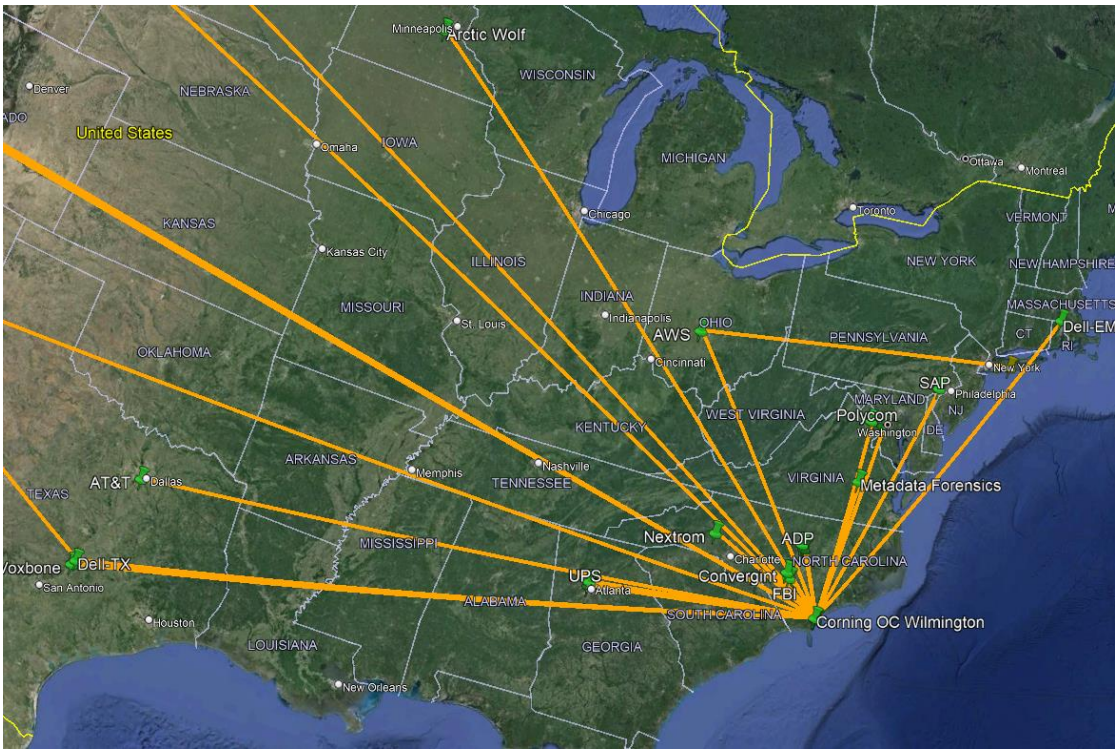
Google Earth's "Places" Explorer, with checkboxes used for view selection:



Legend images for each of the views, to be displayed on the map:

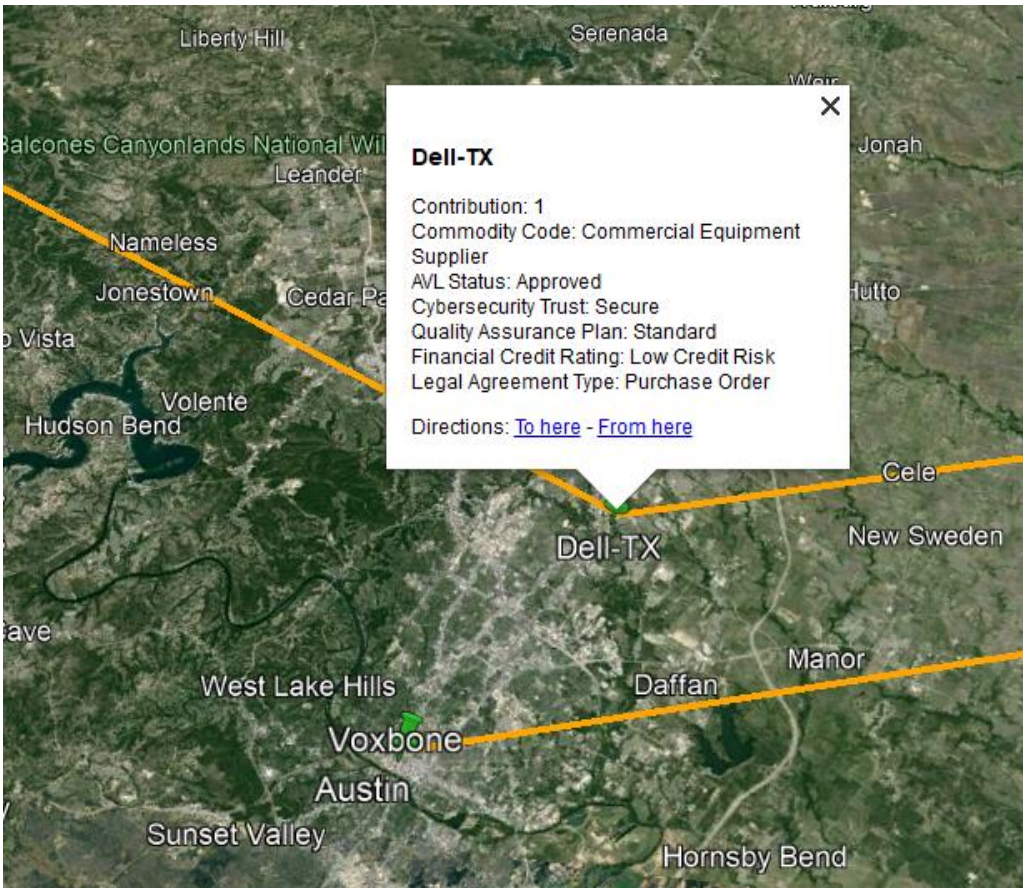


The supply chain map from a distance, showing AVL status:

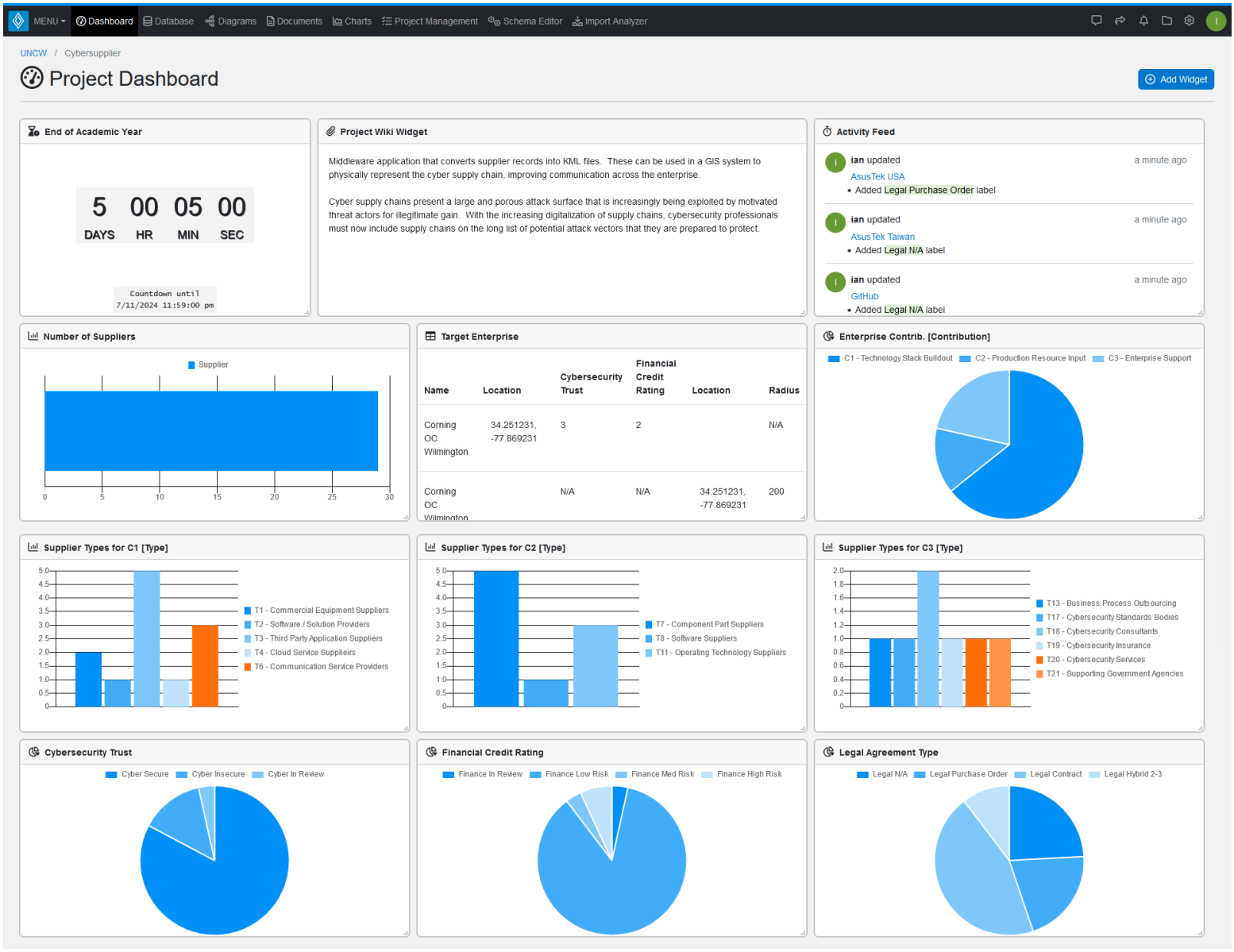


The target enterprise close-up, with green security boundary shown:

The details pop-up of a chosen supplier on the map:



Appendix H
Innoslate Dashboard Widget Samples



Appendix I
Validation Report

Tasks

1: Create a new Supplier with the following attributes. Use the provided code sheet.

Name	NVIDIA
Location	37.370626, -121.969868
Commodity Code	Hardware Suppliers
AVL Status	Approved
Cybersecurity Trust	In Review
Quality Assurance Plan	Standard
Financial Credit Rating	In Review
Legal Agreement Type	Purchase Order

2: Create an Interaction Link from NVIDIA to AsusTek USA.

3: Obtain the total number of suppliers in the map data.

4: Using the Commodity Code folder, find a Cloud Service Supplier on the map.

5: Find a supplier on the map with an AVL designation of "Do Not Use".

6: Find a supplier on the map with a Cyber Trust designation of "In Review".

7: "Dell-TX" is a Cyber Trusted supplier on the map. Does "Dell-TX" have any 3rd party suppliers connected to it? What is their Cyber Trusted status?

8: Update the new supplier's Financial Credit Rating from 1 to 2.

9: Update the new supplier's Cybersecurity Trust from 1 to 3.

10: Delete the new Supplier and the new Interaction Link.

Follow-up questions

1: Was it easy or difficult to create a new Supplier and Interaction Link? Do you have any additional comments?

2: Was the Google Earth interface easy or difficult to navigate while finding supplier information? Do you have any additional comments?

3: Did you find the visual representation provided by Google Earth to be helpful towards improving your understanding of the supply chain?

4: Do you think this system is an effective way to train students by serving as a realistic enterprise model?

5: Do you think this system can improve communication between members of cross functional teams in an enterprise environment?

6: What functionality would you like to see in future versions of this system?

7: Is there anything else you'd like to share about this experience?

Below is the code sheet shown to participants while performing Task 1.

#	Commodity Code
1	Commercial Equipment Suppliers
2	Software / Solution Providers
3	Third Party Application Suppliers
4	Cloud Service Supplieirs
5	System Integrators
6	Communication Service Providers
7	Component Part Suppliers
8	Software Suppliers
9	Hardware Suppliers
10	Service Suppliers
11	Operating Technology Suppliers
12	IoT Device Suppliers
13	Business Process Outsourcing
14	Professional Services
15	Sales Channel Partners
16	Customers
17	Cybersecurity Standards Bodies
18	Cybersecurity Consultants
19	Cybersecurity Insurance
20	Cybersecuriity Services
21	Supporting Government Agencies
ELSE	Other

#	AVL Status
1	Approved
2	Conditional Use
3	Do Not Use

#	Cybersecurity Trust
1	In Review
2	Insecure
3	Secure

#	Quality Assurance Plan
1	In Review
100	Standard
>100	Standard+
>500	Standard++

#	Financial Credit Rating
1	In Review
2	Low Credit Risk
3	Medium Credit Risk
4	High Credit Risk

#	Legal Agreement Type
1	N/A
2	Purchase Order
3	Contract
4	Hybrid 2+3