System Requirements Specification

for

Option 9

Version 1.0 approved

Prepared by Vetronica

Embry-Riddle Aeronautical University

November 21st, 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Everyone	9/29/23	Initial version	Version 1.0

Everyone	10/30/23	Revisions, improvements	Version 2.0
Everyone	11/17/23	Revisions; improvements	Version 3.0
Everyone	11/21/23	Revisions; improvements	Version 3.1

1. Introduction

1.1 Purpose

Team Vetronica (including Michael Hall, Dana Merry, Troy Neubauer, and Zachary Shepley) is developing a cybersecurity planning for executing vulnerability assessment for operational environments, application of cybersecurity in operational environments and conducting decision analysis of crucial infrastructure by implementing model-based systems engineering.

1.2 Document Conventions

The purpose of this document is to give the reader an idea of our requirements for the system we plan on designing as well as the hardware, software, and user expectations. The document is divided into four different categories, each of which cover an essential part of the system.

The Overall Description section of this document will cover the basic requirements of the system, the functionalities, and the overall basic information for the system.

The External Interface Requirements section of this document will cover the interfaces. There are four different interfaces that will be covered: User Interfaces, Hardware Interfaces, Software Interfaces, and Communications Interface.

The System features section of this document will cover the key features pertinent to this system in itself. This includes importing models, analyzing the models, and rendering the results of the analysis.

Lastly, the Other Nonfunctional Requirements category just covers any requirements that were not covered in any of the above sections.

1.3 Intended Audience and Reading Suggestions

The intended audience for this document is project owners of large computer systems which need to work despite cyber threats, as well as anyone interested in MBsE model based cyber security.

1.4 Product Scope

For this Project, the objective is to research and develop a Cybersecurity Analysis tool for use in an Aerospace Technology environment. This tool will be supported by a Model-based Systems Engineering tool to be developed by the team. By the end of the term, the expectation is that there will be a functional Cyber Analysis environment that can be used to identify safeguards in physical

and virtual aerospace environments to protect against the growing cybersecurity threats of today's world.

1.5 References

- Veronica Cyber Analysis Vision Statement
  - https://github.com/Vetronica/CybAnalysisTool/blob/main/Vetronica%20Cyber%20Analysis%20Vision%20Statement.docx
- MagicDraw System Requirements
  - https://docs.nomagic.com/display/MD190/System+requirements
- Capella System Requirements
  - https://github.com/eclipse/capella/blob/master/doc/plugins/org.polarsys.capella.ui.doc/html/Installation%20Guide/How%20to%20install%20Capella%20and%20Addons.mediawiki#table-of-contents
- NIST 800-209
  - https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-209.pdf
- Amazon S3
  - https://aws.amazon.com/pm/serv-s3/?gclid=Cj0KCQiApOyqBhDlARIsAGfnyMpX7V2YDtpd9v8mnsDlN0ffmDUlzBC13LKy7uM6aDXWnmpfEi2w-_caAnNXEALw_wcB&trk=fecf68c9-3874-4ae2-a7ed-72b6d19c8034&sc_channel=ps&ef_id=Cj0KCQiApOyqBhDlARIsAGfnyMpX7V2YDtpd9v8mnsDlN0ffmDUlzBC13LKy7uM6aDXWnmpfEi2w-_caAnNXEALw_wcB:G:s&s_kwcid=AL!4422!3!536324446683!p!!g!!aws%20s3!11204620052!112938567834
- Cybersecurity Data Science (CSDS) Aviation Architecture Framework (AAF) - Utilization Strategy
  - Part 1
    - https://erau.edu/-/media/files/university/research/car/part-1-csds-aaf-utilization-strategy.pdf
  - Part 2
    - https://erau.edu/-/media/files/university/research/car/part-2-csds-aaf-technical-document.pdf
  - Part 3
    - https://erau.edu/-/media/files/university/research/car/part-3-csds-aaf-system-guidance-document.pdf
- White House Special Publication
  - https://www.whitehouse.gov/briefing-room/statements-releases/2023/03/02/fact-sheet-biden-harris-administration-announces-national-cybersecurity-strategy/
- TSA
  - https://www.tsa.gov/news/press/releases/2023/03/07/tsa-issues-new-cybersecurity-requirements-airport-and-aircraft

2. Overall Description

2.1 Product Perspective

This product is a cybersecurity analysis tool targeting aerospace applications via SysML. Using the provided MBSE tools, this will be a self-contained product which will model hardware infrastructure for obtaining a high level view of possible attack vectors.

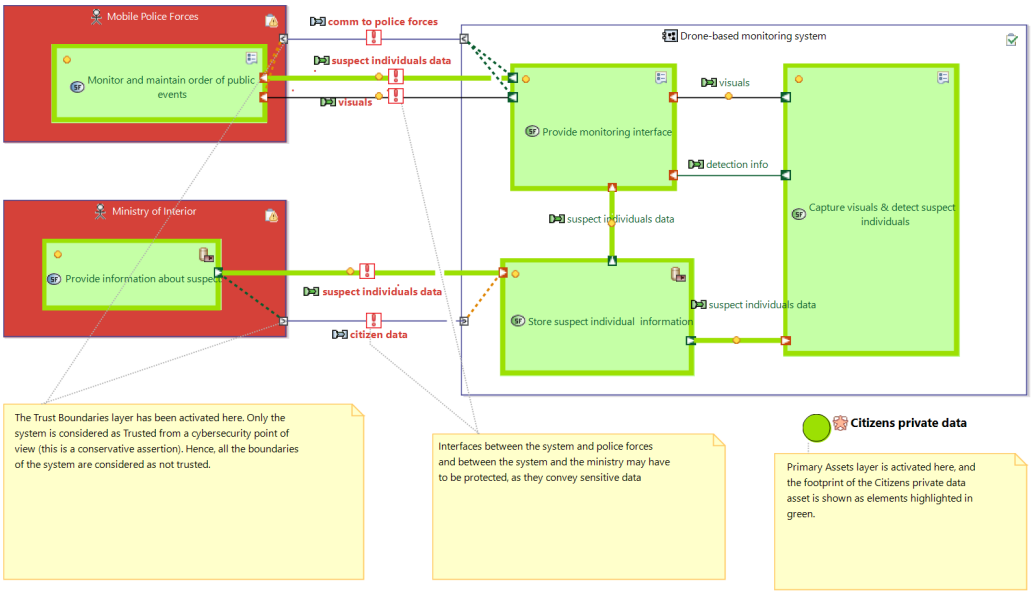

Figure 1: Sample diagram of how Model Based Systems Engineering is conducted in the Capella software

2.2 Product Functions

- Define Network(s) of interest
- Define Threats of concern
- Establish applicable Conceptual data infrastructure and data structure (storage and processing)
- Establish applicable Data sphere
- Establish usage spheres
- Define cybersecurity strategies for specific threats
- Identify Data Acquisition methods (sensors, streaming, security log files)
- Identify current sensor vantage and blind spots (visual schemas)

2.3 User Classes and Characteristics

2.3.1 System Engineers

The System Engineer would be in charge of designing the system in the software, whether it be MagicDraw or Capella. The engineer in charge of this system must have extensive knowledge in SysML and in Model-Based Systems Engineering. This engineer will have access to all MBsE

related functions, designing the system, and preparing the model for analysis for the Cybersecurity Engineers.

2.3.2 Cybersecurity Engineers

The Cybersecurity Engineers would have access to analyzing the results outputted from the system. The engineer must have extensive knowledge in policies related to the aviation field, must know cybersecurity regulations, as well as any other relevant cybersecurity practices. They should have a good knowledge of threats, how to combat these threats, and much more. They would be responsible for creating a Cybersecurity policy to implement whatever results were outputted, and determine whether they follow FAA regulations.

2.4 Operating Environment

This project will be primarily contained in MBSE-related modeling tools. For this project, two different MBSE tools will be used to import models: MagicDraw and Capella. These tools will be used to model a physical environment rather than the allocation of physical resources. At this juncture, the functionality of each tool and scope of integration is unknown.

2.5 Design and Implementation Constraints

Recommendations for running MagicDraw:
2.5.MD1: Must purchase license from Dassault Systems
2.5.MD2: Modern 64 bit CPU
2.5.MD3: 16 GB Ram
2.5.MD4: 1 GB of disk space (This will increase depending on plugins installed).
2.5.MD5: Display Resolution of 1920x1080 or greater
2.5.MD6: Operating System must support Java SE 8

Recommendations for running Capella:
2.5.C1: 64 bit computer with Windows, Macos or Linux. (Linux and Mac support is claimed by Eclipse, but unverified by us)
2.5.C2: An unzipping software such as 7-Zip
2.5.C3: 2 GHz processor
2.5.C4: 4 GB RAM
2.5.C5: 15 GB of available hard disk space

2.6 User Documentation

The following documents make up the required user documentation for the software. This documentation will give the user all of the necessary guidance to work in both the Capella and MagicDraw software.

2.6.1: Capella documentation https://mbse-capella.org/getstarted.html
2.6.2: Capella features https://mbse-capella.org/features.html

2.6.3: MagicDraw documentation https://docs.nomagic.com/display/MD190/User+Guide
2.6.4: MBsE Fundamentals/Tutorial on MagicDraw
https://www.youtube.com/watch?v=t4vRYhEWQOg

2.7 Assumptions and Dependencies

In order to successfully use this software without any complications, the below assumptions are presented to make the experience simple.

2.7.1: Must be working on a device compatible with the recommendations and/or requirements with both MagicDraw/Capella.
2.7.2: Must have skill in usage of MBsE development tools such as Capella and MagicDraw.
2.7.3: Must have an environment to be implemented through the MBsE development tool.
2.7.4: There is a dedicated Systems Engineer and a Cybersecurity Engineer.

3. External Interface Requirements

3.1 User Interfaces

User interfaces will have a similar appeal to MagicDraw. This is due to MagicDraw being widely used, so a software of similar look should be easy to transition.

3.2 Hardware Interfaces

The hardware for this project must be capable of running both the software created here as well as the MBsE software. The interfaces for the hardware can be broken down as follows:

3.2.1: The system is intended to be used on a modern computer running Windows, MacOs or Linux.

3.2.2: The system shall be running an MBsE software such as MagicDraw or Capella.

3.2.3: The system shall have the ability to retrieve the appropriate files from MagicDraw or Capella for analysis.

3.2.3: The system shall have at least 4GB of RAM.

3.2.4: The system shall have at least 16GB of hard disk space.

3.3 Software Interfaces

N/A

3.4 Communications Interfaces

3.4.1: The software will serialize the model state to disk to allow users to save their work as well as distribute finished models.

3.4.2: The software will also check for updates from a centralized server to prompt the user if an update is available.

4. System Features

This section provides a more in-depth list of what was presented in Section 2.2. This includes the functionality on importing models (See Section 4.1), the analysis of the models (See Section 4.2), and the rendering of analysis results (See Section 4.3). This section will go over the requirements for each of these sections.

4.1 Importing of Models

4.1.1 Description and Priority

The user shall be able to import existing models from Capella or MagicDraw into the analysis tool. Separating the import process from business logic allows for flexibility in supporting both formats and improved maintainability.

4.1.2 Stimulus/Response Sequences

The user shall be able to indicate a file / directory save of a Capella or MagicDraw model for import into the analysis tool.

4.1.3 Functional Requirements

4.1.3.1: The software shall allow the user to import MagicDraw model files.
4.1.3.2: The software shall allow the user to import Capella model files.
4.1.3.3: The software shall alert the user if features in a particular save are not supported by the tool.
4.1.3.4: The software shall alert the user if a model is corrupted or cannot be loaded.
4.1.3.5: The software shall display when the model is imported successfully.

4.2 Analysis

4.2.1 Description and Priority

Once a model is imported into the software, the user will have the ability to run high level analysis on the model to uncover possible cyber threats. The exact nature of analysis depends on what concerns the user has. What networks are they interested in? What threats are they most concerned about?

4.2.2 Stimulus/Response Sequences

The user shall indicate the threat models, networks of concern via a config file (for CLI use), or by selecting options in the GUI.
The user shall indicate via a button press or subcommand that they wish to perform analysis.

4.2.3 Functional Requirements

4.2.3.1: The software shall import networks of concern from the user.
4.2.3.2: The software shall import a prioritized list of threats of concern from the user.

4.3 Rendering of Analysis Results

4.3.1 Description and Priority

Once a model is imported into the software, the user will have the ability to run high level analysis on the model to uncover possible cyber threats.

4.3.2 Stimulus/Response Sequences

The software shall display analysis results in a manner suitable for consumption. The exact requirements for this action are currently unknown, as they depend on if we support a CLI or GUI.

4.3.3 Functional Requirements

The software will prepare a render of the results for proper analysis of flaws and vulnerabilities.

4.4 Simulation of Dynamic Systems

4.4.1 Description and Priority

Simulation of dynamic environments is essential to proving correctness and safety of systems. The software will provide the user with the ability to perform conditionals, repeat commands, introduce delays, create state, and modify state (essentially a visual programming language with a minimal scope). This feature will allow the user to simulate the systems they care about and ensure their invariants are upheld.

4.4.2 Stimulus/Response Sequences

Mouse clicks, keyboard input for strings are used by the user to: move programming elements around, connect and order them, rename them, modify values, and delete elements.

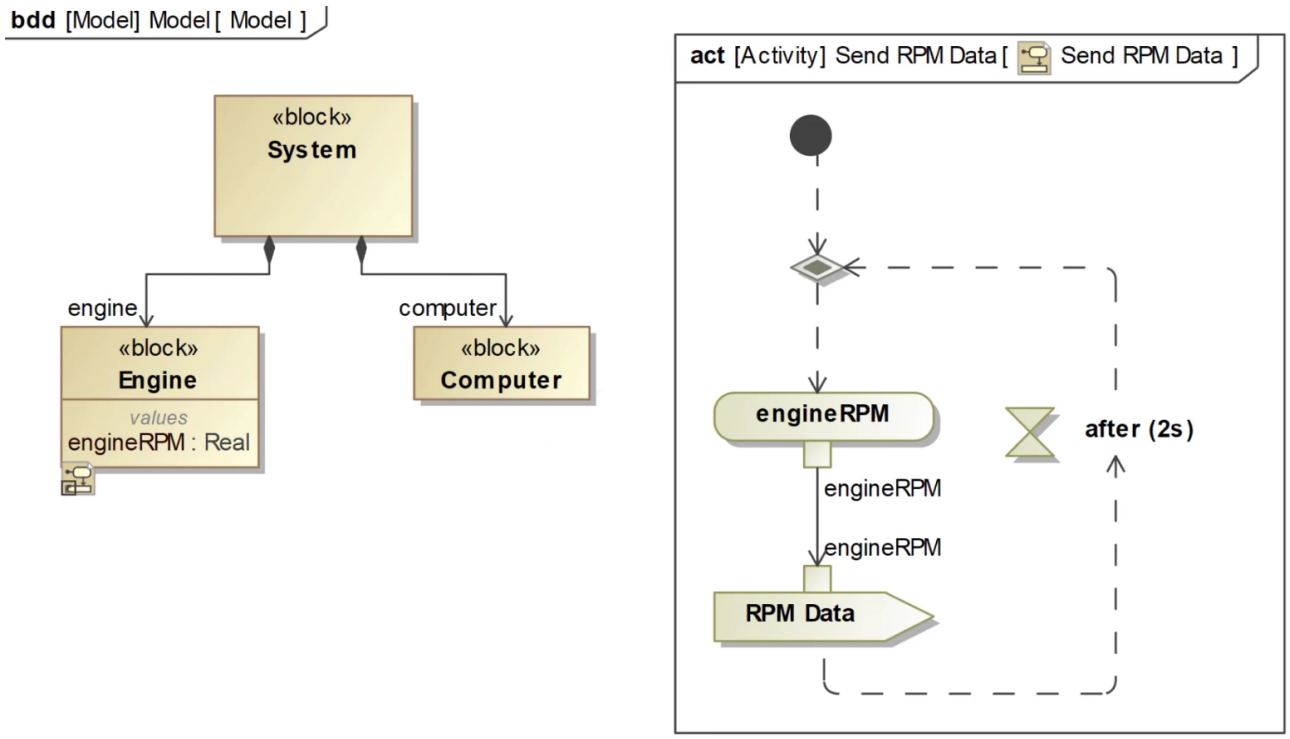

Figure 2: Example of a dynamic simulation in MagicDraw

This program modifies the engine rpm numerical value based on a calculation of its constituent forces, sends this data to the computer, waits two seconds, and repeats from the start unconditionally.

4.4.3 Functional Requirements

4.4.3.1: The software shall store a simulation as a collection of programming elements
4.4.3.2: The software shall run a simulation by executing each programming element in order
4.4.3.3: The software shall respect branches diverging control flow based on standard comparison operators
4.4.3.4: The software shall support loops jumping back to a programming element based on conditional operators
4.4.3.5: The software shall delay execution for a constant or dynamically determined amount of time when the delay programming element is encountered
4.4.3.6: The software shall maintain a global key value pairing of numbers, strings, booleans and objects (element with key value pairs contained inside of it)
4.4.3.7: The software shall assign to the global key value pair when encountering a store programming element
4.4.3.8: The software shall modify number values in the global key pair when encountering a math operation such as addition, subtraction, multiplication, or division acting on two numbers in the global key value pairs

4.5 Plugin System

4.5.1 Description and Priority

Software plugins allow developers to extend functionally without becoming tightly coupled with the internals of the software. Additionally plugins can be licensed differently than core software. For example, several MagicDraw plugins are open sourced under GPL or MIT while the core MagicDraw software by Dassault Systèmes is proprietary. Open sourcing plugins can grow a community around plugins and utilize the power of open source to implement feature requests and fix bugs more quickly.

4.5.2 Stimulus/Response Sequences

When the software is opened, the list of registered plugins are loaded. Users can select additional plugins to register via an option in the toolbar.

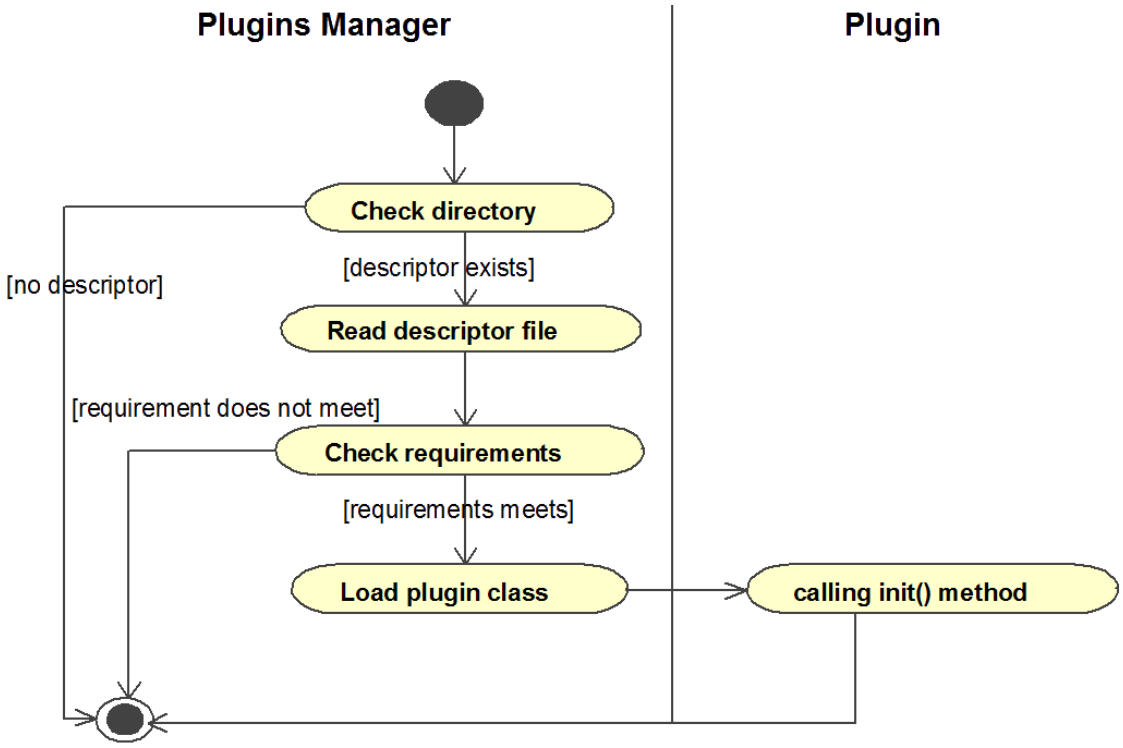

Figure 3: Plugin architecture of MagicDraw

Our proposed plugin system is simpler given only a single file shared object is needed to contain the plugin functionally and communicate version requirements. Additionally the requirement that plugins are compiled to native code in a shared object file will reduce memory consumption and likely lead to a faster user experience given that plugin functionally will not require the use of a JVM.

4.5.3 Functional Requirements

4.5.3.1: The software shall load plugins as shared object files which export required c abi functions
4.5.3.2: The software shall store a list of paths to registered plugins
4.5.3.3: On startup the software shall attempt to load each registered plugin from the list
4.5.3.4: Each plugin will export a function named `plugin_init` which takes a pointer to a struct containing functions that can be called by the plugin to interact with the core software
4.5.3.5: The plugin will export a function called `plugin_version` that returns an int containing the plugin's version and required version of the software
4.5.3.6: The software shall reject plugins which return an error code from `plugin_init` or require a future version of the software

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- A data warehouse should address the following characteristics:
    - Determine necessary capacity and scaling plans
    - Ensure granular metadata and object versioning
    - Determine the average and peak number of requests to the API and any desired query processing
    - Identify the frequency and size of data transfer (ingress and egress)
- Object storage systems are recommended because they are easily scalable, support granular metadata, and are best suited for relatively static data storage
- Generally, the speed of storage media is proportional to the cost
- The number of users accessing the data warehouse and the frequency of those accesses will have an impact on the system's ability to address each request in a timely manner

5.2 Safety Requirements

5.2.1 Scope of Product Safety

As this product is largely contained in a virtualized environment, the primary objective when referring to safety lies in ensuring the integrity of the software and systems the product implements. In order to ensure the safety of the system, the following focus areas should be addressed:

5.2.1.1 Malware/Ransomware

One of the most significant factors in ensuring the integrity of a system is ensuring that the system remains healthy and free of malware. Malware can cause significant and in some cases irreversible damage to a computer. There are countless avenues for malware to infiltrate a system, most notably through tactics such as social engineering. Once inside a system, malicious files and programs can

perform a variety of data collection and corruption techniques. For instance, ransomware can be used to encrypt all of the data on a system and require an organization to pay a cost or "ransom" to unencrypt it. By ensuring that all systems and software are running an advanced antivirus and firewall protocol, the risks of malware can be significantly mitigated.

5.2.1.2 Deliberate Misconfiguration/Human Error

The most significant concern for any security environment is the impact of human performance of the safety and integrity of the system. Overwhelmingly, human error is the largest source of risk to an organization's or product's assets. Errors such as typos, miscommunications, succumbing to social engineering, and the lack of knowledge or familiarity with vendor best practices are common types of human errors in an organization. While many of these threat vectors can seem menial and insignificant, they can lead to catastrophic consequences for an organization and invite a host of vulnerabilities into a system. For instance, an inadvertent click of a button from a system administrator could lead to a restricted object storage pool being exposed to a public network. While many of these errors can be accidental in nature, there is also potential for an insider threat performing a malicious and deliberate misconfiguration of data within the system. This type of threat can be more difficult to prevent, but through training and awareness, employees can be taught the typical signs of a potential insider threat to prevent or mitigate damages.

5.2.1.3 Data Breach/Data Exposure

A data breach is an incident that involves classified or otherwise sensitive data escaping the scope of its security. Common data breaches include PII, PHI, credit card information, or system passwords. What makes data breaches and exposure dangerous is the opportunity for threat actors to access leaked information and use it to infiltrate or otherwise damage the reputation of the product and the organization. Data breaches can be exposed in a public setting, but they can also be leaked outside of their scope but within the organization's systems. Internal data leaks often lead to less damage from an organizational standpoint, but can often introduce the threat of insider threats and other damaging misconfigurations as referenced in Section 5.2.1.2.

5.2.1.4 Data Corruption

Data corruption occurs when damage or errors in a system during writing, reading or other storage processes result in unwanted changes to a data set. Most often, when data corruption occurs, the data will produce unexpected and incorrect results. If a data or document file as a whole is corrupted, it can lead to loss of access or critical errors resulting in a loss of data as a whole. Malware can often be a contributor to data corruption, as well as human error or misconfiguration. Corruption more often than not leads to significant damages to an organization's well-being, especially if there is not a sufficient backup system in place. Therefore, it is important to ensure that any software systems have countermeasures and safeguards in place to mitigate the impact of potential data corruption.

5.2.1.5 Denial of Service

DoS occurs when a user is unable to access their data or systems. This could occur for a variety of reasons, most commonly from malware or other forms of data corruption. By denying employees access to systems, the overall operation of the organization is hindered significantly and often eliminated until such time that the current or backup version can be brought online. While the effects of DoS are often reversible, the most damaging aspect is the cost incurred from the systems being offline. By taking down an organization's systems, an attacker can remove that organization's ability to operate, perform necessary tasks, and generate income.

5.2.1.6 Compromising Backups

Backups are essential to the effective maintenance and operation of an organization. By maintaining updated backups, an organization can limit damage, downtime, and financial losses by bringing systems back online and restoring functionality quickly and efficiently. While there are several implementations for backups based on price and downtime requirements (i.e. Hot/cold/warm sites, full vs. partial backups, required downtime percentage), the specific implementation will vary by organization based on budgets, time, and manpower leveraged. However, it is absolutely essential that every functional organization in a computing environment has a DRP in place that includes written policy for maintaining and deploying backups.

5.3 Security Requirements

5.3.1 Importance of Security

In order for a product to be functional and permitted to be used in an aerospace environment, it must meet the required specifications of all government and industry-related standards. These standards include, but are not limited to, TSA, NIST, and The White House.

5.3.2 Applicable Regulations

5.3.2.1 TSA

In response to the growing concern regarding advanced persistent threats against critical infrastructure in the aviation sector, the Transportation Security Administration (TSA) has issued new guidance. The TSA issued an emergency amendment to regulate the cybersecurity requirements of aviation infrastructure and equipment. This amendment requires that all entities regulated by the TSA to develop and submit an approved implementation plan that outlines their process for maintaining security and availability of their systems, and to prevent degradation and destruction of their infrastructure. In addition, the following actions must be implemented:

1. Develop network segmentation policies and controls
2. Create access control measures to secure and prevent unauthorized access to critical cyber systems
3. Implement continuous monitoring and detection policies and procedures to defend against, detect, and respond to cybersecurity threats
4. Reduce the risk of exploitation of unpatched systems on critical cyber systems in a timely manner using a risk-based methodology

5.3.2.2 NIST

The National Institute of Standards and Technology (NIST) provides governing documents and policies for cybersecurity and analysis. One of these publications, NIST SP 800-209, Security Guidelines for Storage Infrastructure, provides a guideline for the proper implementation and management of storage information and systems. The primary purpose of this document is to provide a comprehensive set of security recommendations for the current landscape of the storage infrastructure. Building an effective risk management program for storage infrastructure based on the security controls described in this document and tightly integrating it with existing cybersecurity frameworks could significantly improve an organization's resilience to different kinds of attacks on data resources.

5.3.2.3 The White House 2023 Cybersecurity Strategy

In a press release from the White House in March of 2023, the National Cybersecurity Strategy was published to guide the nation and their approach to cyberspace. The goal of this new strategy is ultimately to reimagine cyberspace as a tool to achieve the nation's goals in a way that reflects its values. To better realize this vision, it is recommended that the nation make a couple key shifts when approaching cyberspace:

- The burden of cybersecurity must be shifted away from individuals, small businesses, and local governments, and onto the organizations that are most capable and best-positioned to reduce the nation's risks
- Incentives must be realigned to favor long-term by balancing defense and strategic planning to invest in a resilient future

This Strategy aims to enhance collaboration in cyberspace around five pillars:

1. Defend Critical Infrastructure
    a. Expand the use of minimum cybersecurity requirements in critical sectors
    b. Enable public-private collaboration
    c. Defend and modernize Federal networks
    d. Update Federal incident response policies
2. Disrupt and Dismantle Threat Actors
    a. Disrupt adversaries by employing tools of national power
    b. Utilize scalable mechanisms and employ the private sector to perform disruption activities
    c. Align with international partners in addressing the threat of ransomware through a comprehensive Federal approach
3. Shape Market Forces to Drive Security and Resilience
    a. Promote privacy and security of personal data
    b. Shift liability for software products and services
    c. Promote investments in new infrastructure through Federal grant programs
4. Invest in a Resilient Future
    a. Reduce systemic technical vulnerabilities in the foundation of the Internet and across the digital ecosystem
    b. Prioritize cybersecurity research and development for next-generation technologies
    c. Develop a diverse and robust national workforce

5. Forge International Partnerships to Pursue Shared Goals
    a. Leverage international coalitions and partnerships among like-minded nations to counter threats to the digital ecosystem
    b. Increase the capacity of international partners to defend themselves against cyber threats
    c. Create secure, reliable, and trustworthy global supply chains for information and communications technology and operational technology products and services

5.4 Software Quality Attributes

Any software developed in this product will be modeled based on inspiration from MBsE tools such as MagicDraw and Capella. In addition, the quality of the software must be in accordance with applicable Federal policy regarding data construction and implementation.

5.5 Business Rules

The business rules for this product require that all systems, software, and implementation follow all functional and nonfunctional requirements as defined in this document. In addition, any system implementing aviation related information must be designed to incorporate any and all relevant cybersecurity and storage-related policy, procedure, or best practice.

6. Other Requirements

Appendix A: Glossary

CLI - Command Line Interface
GUI - Graphical User Interface
MBsE - Model Based systems Engineering
Rust - The Rust programming language: A language empowering everyone to build reliable and efficient software.
STPA - Systems Theoretic Process Analysis
TSA: Transportation Security Administration
NIST: National Institute of Standards and Technology
PII: Personally Identifiable Information
PHI: Personal Health Information
DoS: Denial of Service
DRP: Disaster Recovery Plan

Appendix B: Analysis Models

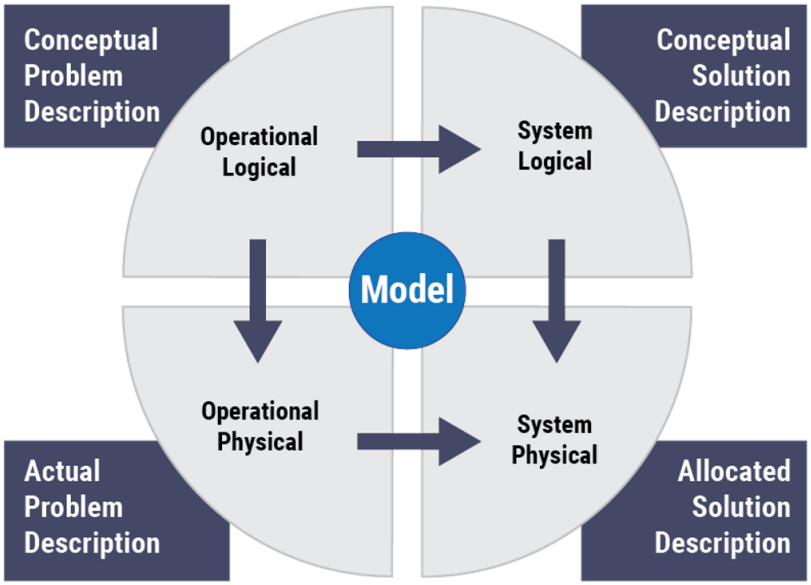

Figure 4: An ideal functionality model for MBsE tools

Appendix C: To Be Determined List

N/A