System Design Document

For

Option 9

Team members: Michael Hall, Zachary Shepley, Troy Neubauer, Dana Merry

Version/Author	Date
Version 1.0; Everyone	9/29/2023
Version 2.0; Everyone	10/30/2023
Version 3.0; Everyone	11/17/2023
Version 3.1; Everyone	11/21/2023

TABLE OF CONTENTS

SYSTEM DESIGN DOCUMENT

1 INTRODUCTION

1.1 Purpose and Scope

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. 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.

1.2 Project Executive Summary

This section provides an overview of the Cybersecurity Analysis project from a management perspective, showing the two Model Based Systems Engineering softwares from which this project will be working from.

1.2.1 System Overview

The goal of MBsE software tools MagicDraw and Capella is to provide a tool for engineers to have a complete MBsE workspace. It also allows for the user to have the resources and plugins needed to successfully design architecture for the necessary systems. For the Cybersecurity Analysis project the softwares will be used to design models, and the goal is to use both MBsE softwares to create the perfect software for MBsE integration of a Cybersecurity Analysis Tool for usage within a data warehouse.

The below figure is an example of how these MBsE tools work toward designing. This shows the process of an Amazon S3 lifecycle. The diagram breaks down the many different policies and shows the overall architecture flow.

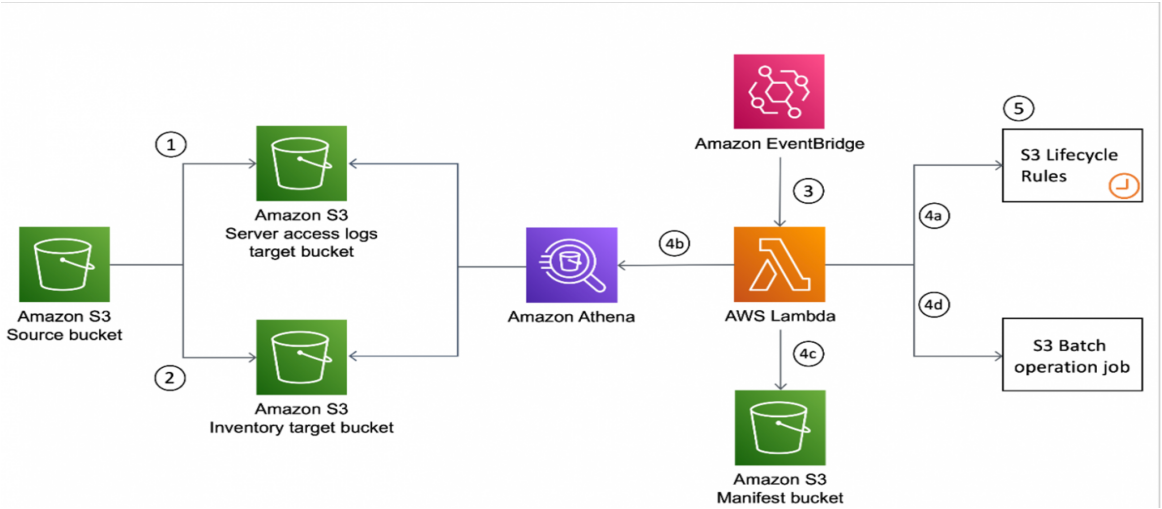

Figure 1: A diagram of an S3 Lifecycle in an Amazon data warehouse

1.2.2 Design Constraints

These tools have a couple major implications when considering the features.

One implication for example is “cross-platform”. Both tools have different features and different plugins. The constraints that designers will be dealing with is the inability to cross reference different plugins between softwares. For example if a user were to prefer the overall interface of the Capella tool but found the Cameo Safety and Reliability Analyzer Plugin used in MagicDraw to be useful, they would not be able to use both.

The second implication is the financial limitations. Capella is a free open-source MBsE software whereas MagicDraw is a software that the users must pay for. The cost of MagicDraw is very expensive and companies who may be on a budget or are looking for free software will only be limited to the capabilities and plugins of Capella.

1.2.3 Future Contingencies

The software shall have plugins from both MagicDraw and Capella to be able to create the ideal product for the Cybersecurity Analysis Tool to analyze. The software shall be able to import files from both MagicDraw and Capella. The ideal goal of this software is to take all of the good from MagicDraw and all of the good from Capella and have one good software for MBsE.

1.3 Document Organization

The design of this document is to give the reader a small idea of the design of MagicDraw and Capella, both have similar capabilities, although they differ in some aspects. The following sections will provide information on the softwares’ capabilities, functions, interactions, limitations, and design.

1.4 Project References

This section provides a bibliography of key project references and deliverables that have been produced before this point.

Our github page can be found at:
https://github.com/Vetronica/CybAnalysisTool/tree/main

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

MagicDraw System Requirements:
https://docs.nomagic.com/display/MD190/System+requirements

Google Cloud Information:
https://cloud.google.com/learn/what-is-cloud-storage#:~:text=Cloud%20Storage%20uses%20remote%20servers,machine%20on%20a%20physical%20server.

Microsoft Azure Information:
https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/get-started/what-is-azure

Amazon S3 Information:
https://www.ibm.com/docs/en/app-connect/cloud?topic=apps-amazon-s3
https://aws.amazon.com/blogs/architecture/expiring-amazon-s3-objects-based-on-last-accessed-date-to-decrease-costs/

1.5 Glossary

CLI - Command Line Interface
GUI - Graphical User Interface
MBsE - Model Based systems Engineering

2 SYSTEM ARCHITECTURE

2.1 System Hardware Architecture

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

2.1.MD6: Operating System must support Java SE 8

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

2.2 System Software Architecture

Capella has its diagrams split into about 4 sections, regarding each of MBsE. Transitioning to the MagicDraw software, it is broken down into three different screens. The figure below (Figure 3) displays a screenshot of this software. The top left screen is where users can view all of the current packages and diagrams as well as create different diagrams to add to the different packages. MagicDraw supports many different diagrams, some examples being Block Definition Diagram, State Machine Diagram, and many more. This category also allows you to sort the information by diagrams and by the overall structure. The bottom left screen gives a small preview of what the diagrams look like without any text. This also gives you three views, “Zoom”, “Documentation”, and “Properties”. The final screen is the screen where users will be working the most with. There are multiple tabs that allow easy switching between diagrams, and the box on the left allows easy adding of elements for whichever diagram the user is working in. The architecture of the software is overall very user friendly.

Having a set up like that, combined with MagicDraws ability to be more centralized would be ideal for a MBsE tool for modeling systems. To elaborate, giving the Capella interface (see Figure 2) and combining it with MagicDraw’s overall user accessibility and interface (see Figure 3) would make the ideal MBsE tool.

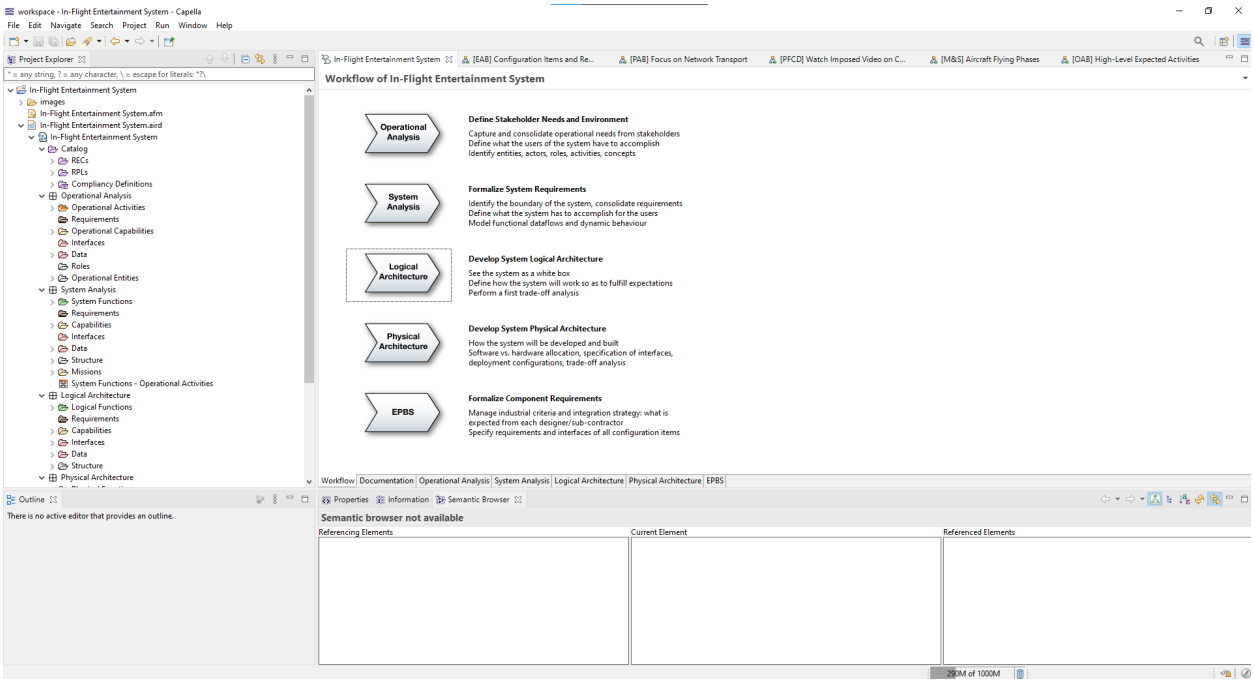

Figure 2: The MBsE interface of all 4 parts of MBsE

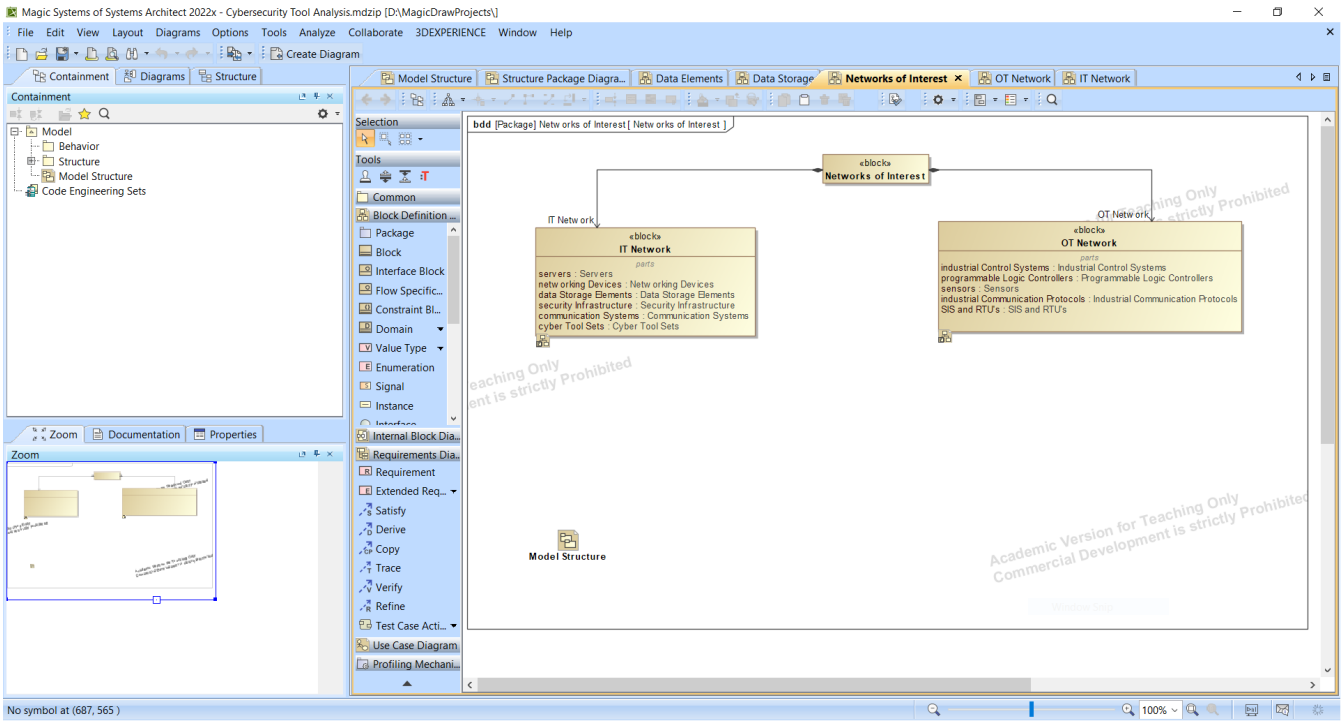

Figure 3: The software architecture of MagicDraw showcasing all of the features provided

2.3 Internal Communications Architecture

N/A.

3 HUMAN-MACHINE INTERFACE

This section provides the design of the systems and the inputs and outputs that will relate to the user of the software. The inputs will relate to how information is added to the system and the outputs will refer to the results that are given back to the user.

3.1 Inputs

The inputs for MagicDraw include lots of key information for it to carry out the design of the software architecture. Requirements will be added into the software using the requirements diagram given by MagicDraw, where the user will be required to input all relevant requirements. Users will also be required to input package names, block names, and relevant information related to the appropriate diagrams that one may design.

Capella as well involves the user having to pick and select projects, filling out the 4 main parts of MBsE: The Operational logic, System logic, System physical, and Operation physical. Requirements are made and added by the user, charts, boxes, and users can all be added into Capella. Users, names, and logic that occurs would all be needed to be inputted by the user.

The ideal MBsE tool would work similarly to both softwares where user input is concerned, as neither tool contains any features that would separate itself from the latter tool.

3.2 Outputs

The outputs for MagicDraw include being able to display results of the model that the user is working on. For example when working with multiple packages in a Package Diagram, users can have the software display the charts by their relations with one another, and as a result the software will give the output of the packages and how they are related to one another. Figure 4 displays an example of how this looks in action.

The outputs for Capella would be depending on which form the user has decided to view, outputs can range from written documentation to pictures such as Figure 5. Notice the multiple tabs with each being a different function of the same system.

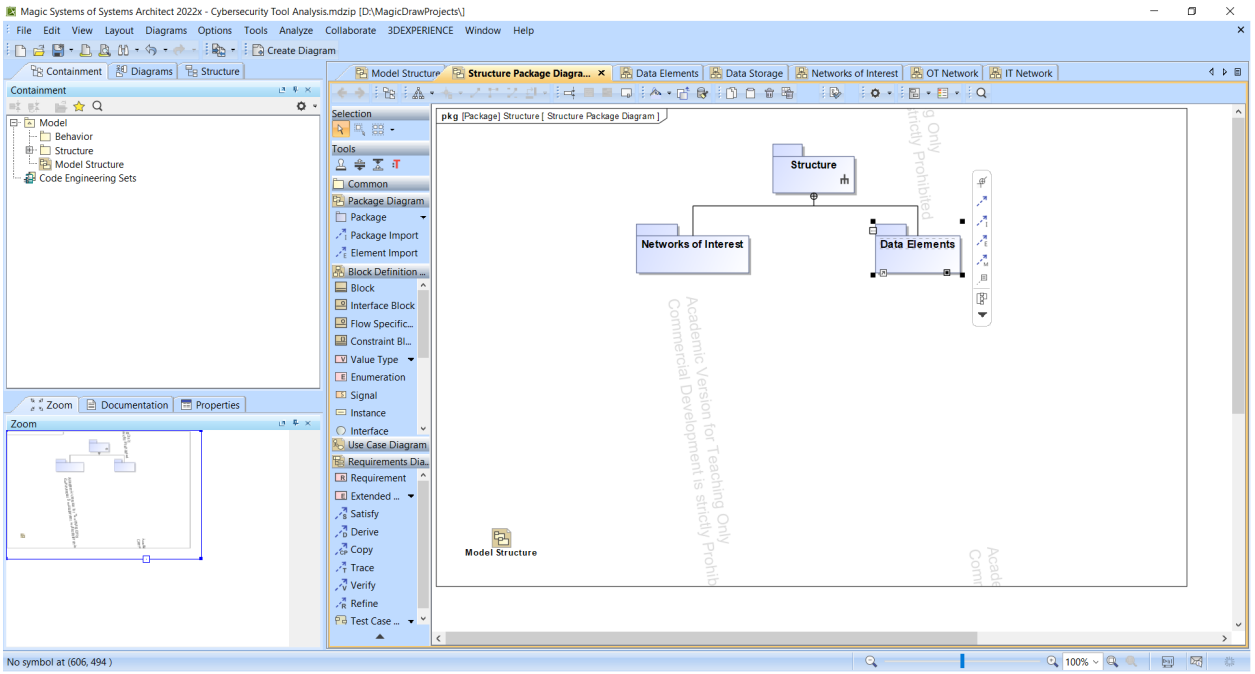

Figure 4: A brief preview of how some outputs would look in MagicDraw

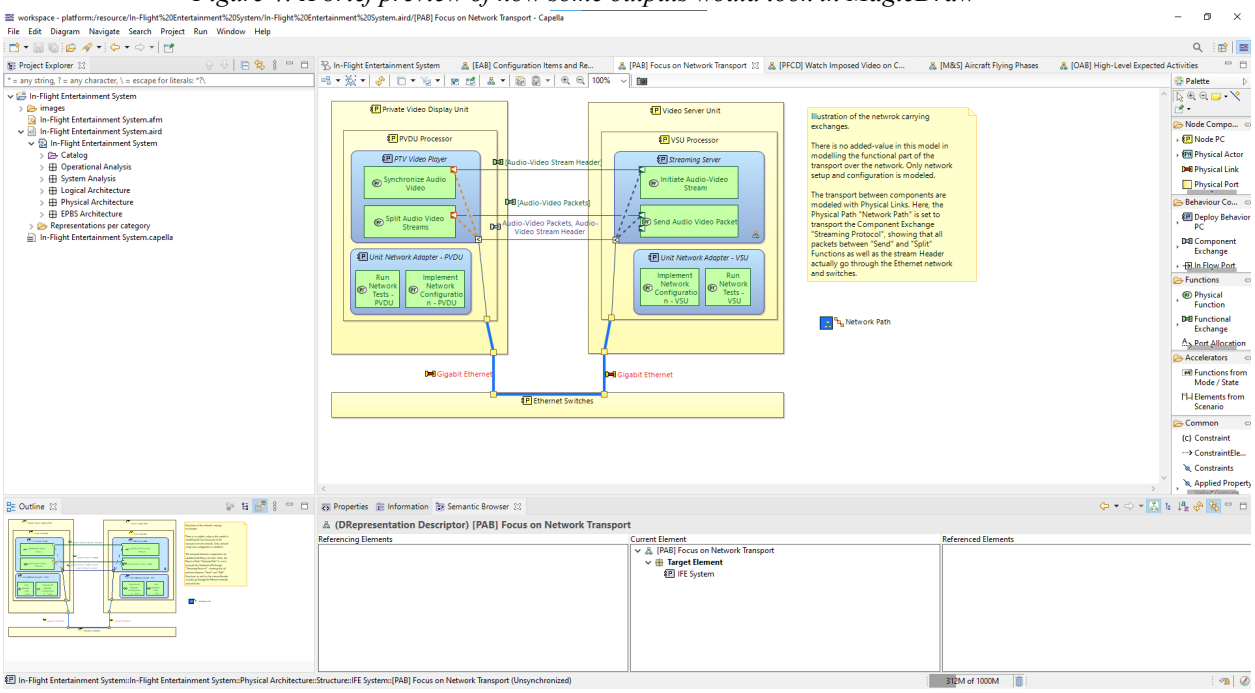

Figure 5: A brief preview of how some outputs would look in Capella

4 DETAILED DESIGN

This section provides the information needed for a system development team to actually build and integrate the hardware components, code and integrate the software modules, and interconnect the hardware and software segments into a functional product. Additionally,

this section addresses the detailed procedures for combining separate COTS packages into a single system. Every detailed requirement should map back to the FRD, and the mapping should be presented in an update to the RTM and include the RTM as an appendix to this design document.

4.1 Hardware Detailed Design

Hardware is provided by the user and we aim to maximize hardware compatibility with our software. For an ideal software it is unlikely it would have strict requirements as the only matter which would affect them is time requirements. For instance a user that is running an older system may have a slower experience, however they shall be able to access all necessary features and plugins needed to ensure full functionality.
Outside of the user's computer, there are no real hardware components needed for the softwares in this document.

4.2 Software Detailed Design

The intended design of the ideal MBsE tool works as follows. The initial startup menu would look as Capella's menu does (see Figure 2). Once the user has gotten off the startup menu and proceeds to development and structuring diagrams, it would look similar to the MagicDraw interface (see Figures 3 and 4). The overall user interface and functionality would also be based on MagicDraw as shown in those figures.

This ideal software would contain plugins from both MagicDraw and Capella.

From the MagicDraw side, it would contain the Cameo Safety and Reliability Analyzer plugin. This plugin gives risk management analysis, and while this plugin was initially used for medical devices, it is applicable to all other devices and architectures also. Some features of this plugin include ensured traceability, performance of safety analysis, and increased agility between risk analysis and design.

The next plugin would be the Cameo Requirements Modeler Plugin. This plugin comes with features which are very beneficial within the MBsE scope. Its features include importing/exporting ReqIF files to utilize them with other tools, easy traceability between different models and architectures, tracking of metrics, and it also allows the user to create their own requirement types to increase the functionality of the plugin to their liking.

From the Capella side, it would contain the STPA plugin. STPA is a plugin that focuses on safety but is adaptable to the cybersecurity field. This tool will enable system architects, designers, domain experts, safety or risk analysis experts to perform STPA analysis in model-based fashion. This plugin is an experimental one however it has been used successfully in many real-world projects.

The next Capella plugin would be the Darc Viewpoint plugin. This plugin provides cybersecurity assets and threat modeling analysis. This plugin is intended to support co-engineering between system and cybersecurity engineering teams when performing tasks. Those tasks include but are not limited to identifying assets and trust boundaries, identifying threats, and characterizing security needs.

With all of these plugins and features from both softwares, this would make the ideal MBsE tool for Cybersecurity Analysis.

4.3 Internal Communications Detailed Design

As we will be distributing a statically linked binary, the only communications our application will have are:

- Input of model file from Capella or MagicDraw into application memory via the file system
- Input of networks of interest, attack vectors user cares about, etc. to inform conditional analysis options
- Internal data flow via functions and shared state to implement analysis business logic
- Output results of analysis (initially printed to console)

5 EXTERNAL INTERFACES

This section will discuss the external interfaces associated with this Cybersecurity Analysis tool.

5.1 Interface Architecture

While there are no specific architecture implementations currently defined in this product, there are several industry-standard architectures that can be used to implement its design, including:

- Amazon S3
    - Implements IBM in both virtualized and physical environments.
- Google Cloud
    - Uses remote servers for saving data such as files, business data, videos, or images.
- Microsoft Azure
    - Is a huge collection of servers and networking hardware which orchestrates configuration and operation of virtualized hardware and software.

5.2 Interface Detailed Design

As referenced in Section 5.1, specific architecture and interface implementation will be dependent on the environment used to house the product. As a result, specific design processes will utilize best practices and tools provided by the hosting environment.

6 SYSTEM INTEGRITY CONTROLS

Unauthorized access to analysis results showing possible vulnerabilities in the system could be critical to a user if accessed by a threat actor. Specific protections against unauthorized access are best left to system administrators since this configuration is outside the scope of our analysis tool and an unsafe network will compromise confidential data despite any attempts within the analysis tool to keep models or analysis data private.