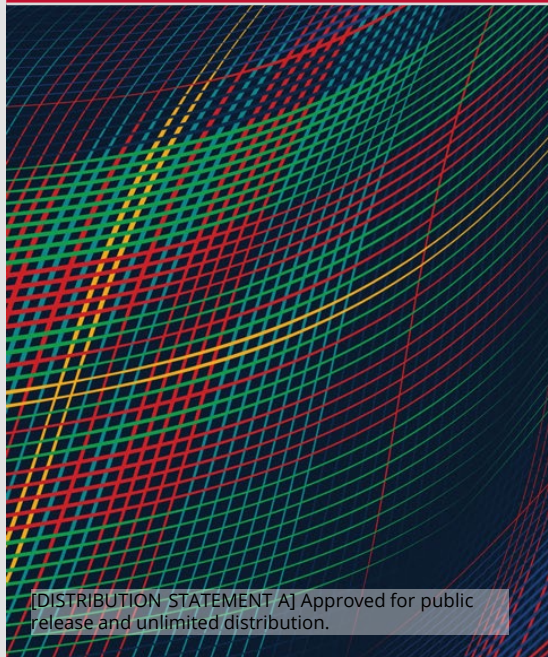
Proactive Architectural Analysis of Cybersecurity Threats

AUG 19 2025

Alexander Vesey
Software Engineer

Natahsa Shevchenko
Sr. Engineer



Document Markings

This material is based upon work funded and supported by the Department of Defense under Air Force Contract Nos. FA8702-15-D-0002, and FA870225DB003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM25-1061

DoD contract number FA8702-15-D-0002

Agenda

- What is MBSE?
- Can I use a whiteboard to hunt threats?
- Security analysis and modeling throughout the lifecycle
- Threat Modeling with MBSE
- PAACT-BUS Step by Step

What is MBSE?

Model-based systems engineering (MBSE) is a formalized methodology that is used to support the requirements, design, analysis, verification, and validation associated with the development of complex systems.

MBSE is:

- Standard modeling language AND
- Digital modeling environment AND
- Modified SE processes AND
- Models built for a purpose AND
- Portfolio of executable models AND
- …

MBSE is NOT:

- About diagrams
- ONLY about modeling tools
- ONLY a modeling language (UML, SysML, UAFML, …)
- JUST an enhanced traditional System Engineering
- An aid for a document-based process

It's NOT a quick fix! But it can become a game changer, if implemented right …

Is Diagram Enough?

I have an idea!

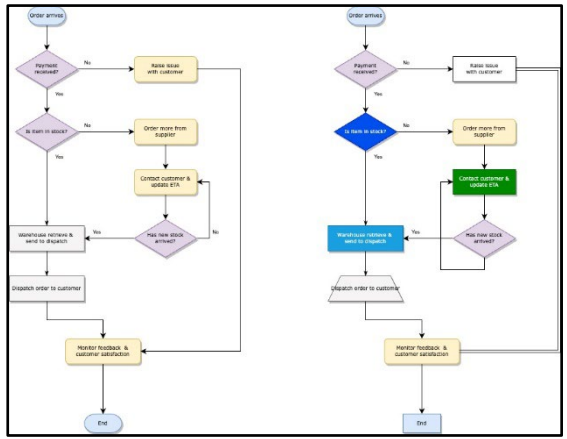
Let's communicate technical details

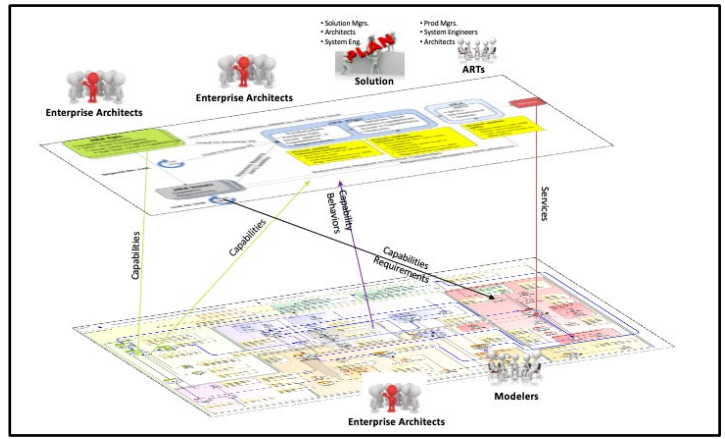
I need to perform analysis

Sketch

Diagram

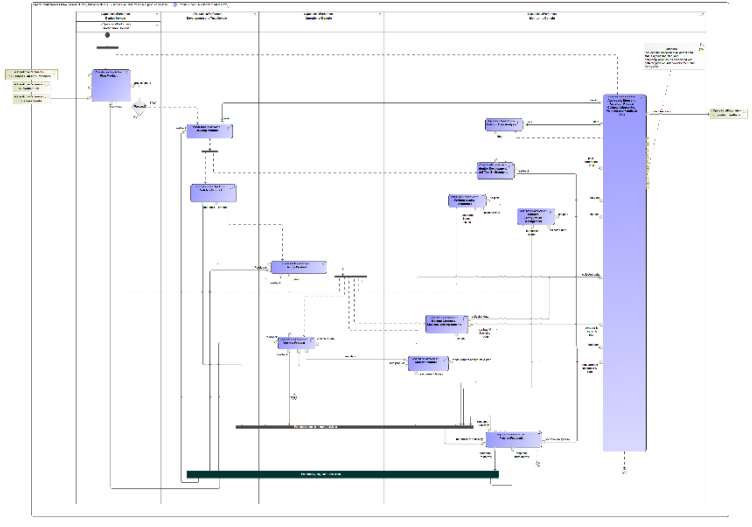
Model Views

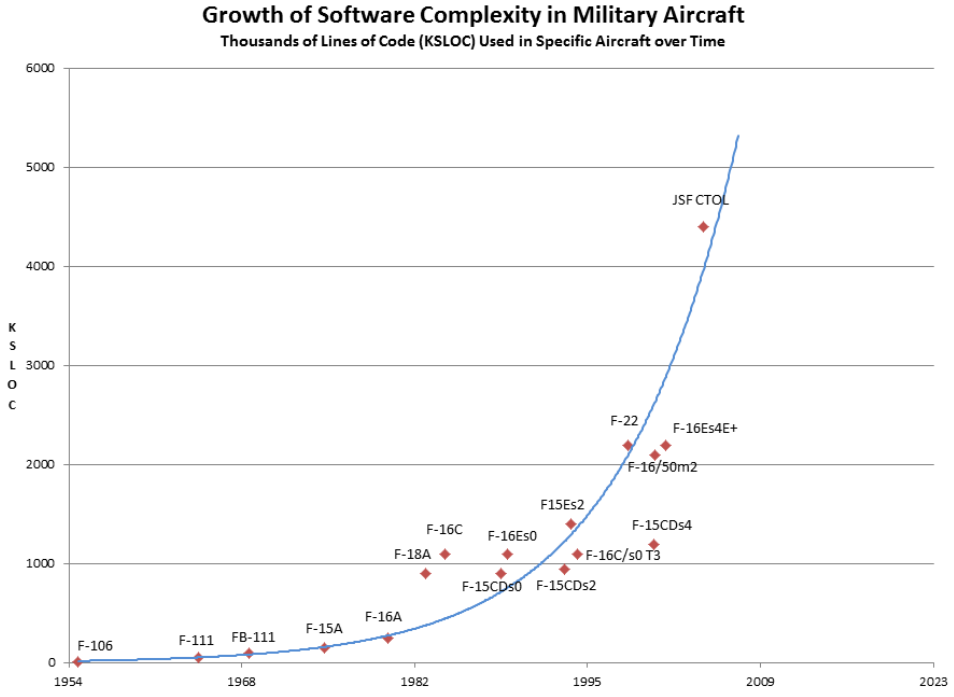




System Size and Complexity

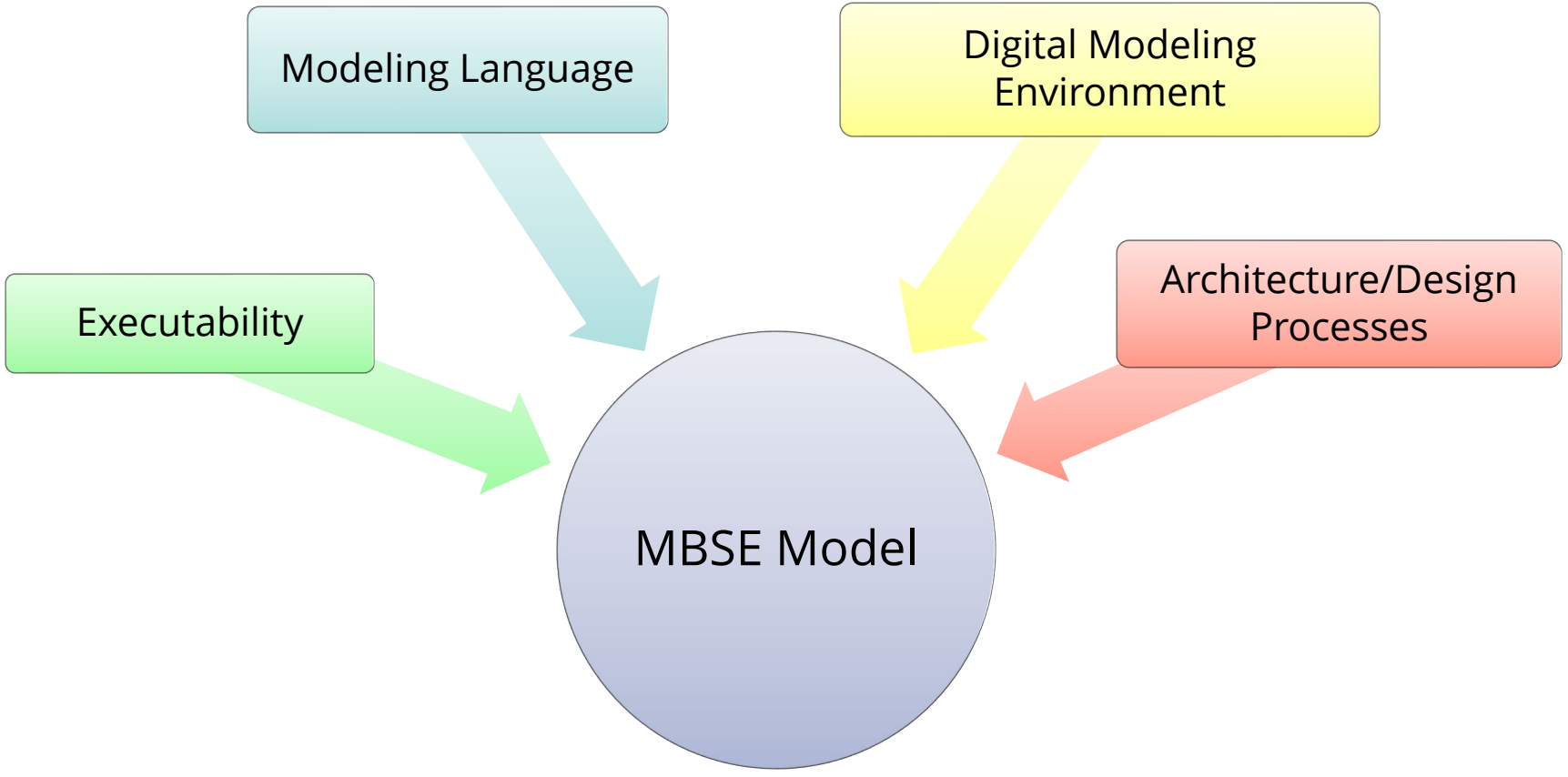
Beyond a certain number of system components, the ability to manage the interfaces connections and data flows necessitates a model.



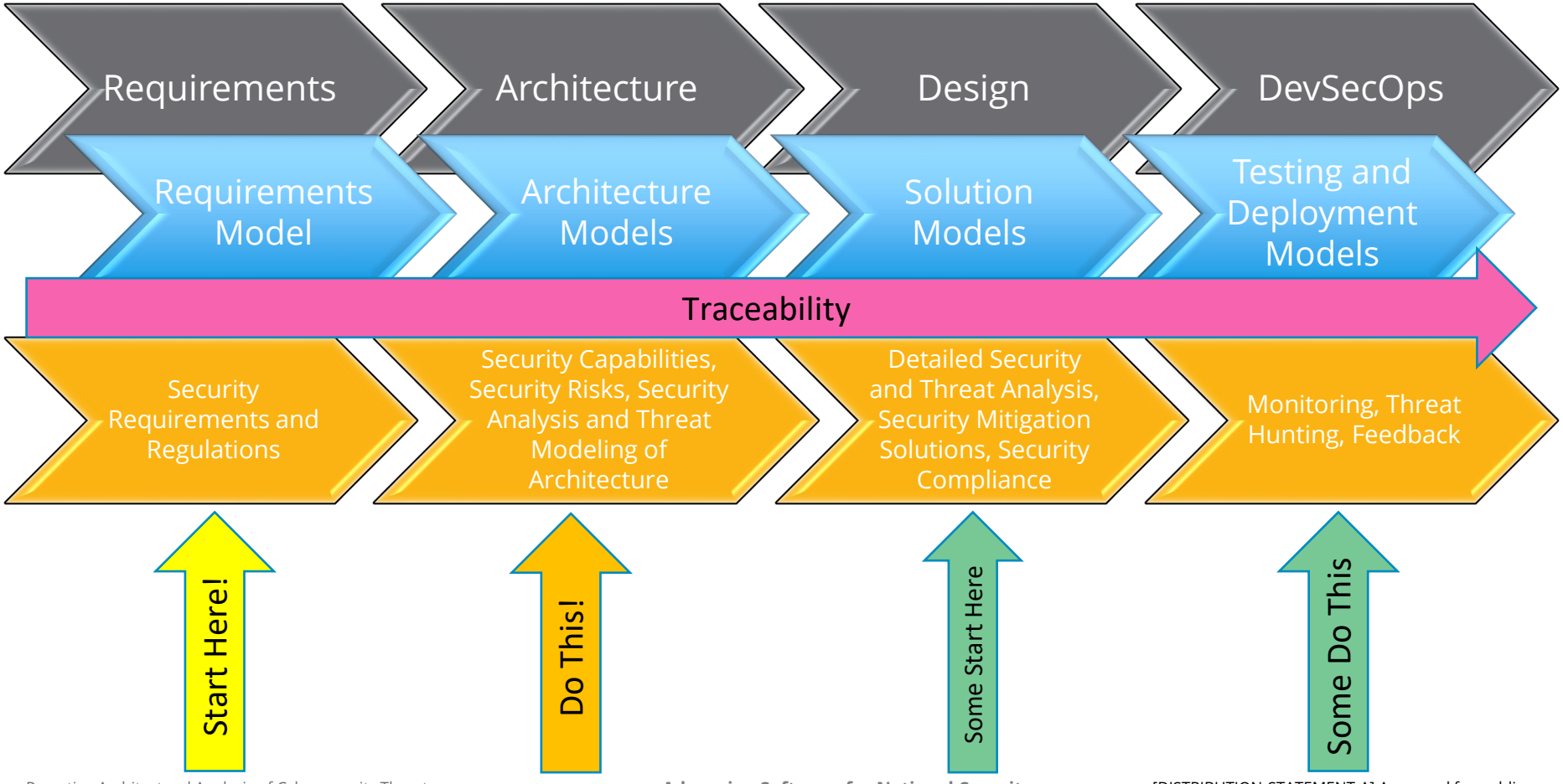


Graphic from Aerospace Vehicle Systems Institute [3]

What is MBSE Model?



SE Life Cycle and Security Analysis



Proactive Architectural Analysis of Cybersecurity Threats

Threat Modeling with MBSE

Carnegie Mellon University
Software Engineering Institute

Threat Modeling Process As Part of Security Analysis

1. What are we building? → System model
2. What can go wrong? → Threat scenarios
3. What should we do about these wrongs? → Security controls
4. Did we do a good enough job of analysis? → Analyze the model

PAACT-BUS: Proactive Architectural Analysis of Cybersecurity Threats – Based on UAF and STRIDE

PAACT-BUS is a comprehensive methodology that integrates a widely adopted architectural framework with a mature threat modeling method to effectively identify, contextualize, analyze, and mitigate cybersecurity threats in software-intensive systems.

By focusing on high-level architectures and threat modeling early in the system development lifecycle, PAACT-BUS ensures that potential vulnerabilities are addressed before specific technologies are finalized. The methodology utilizes the Unified Architecture Framework (UAF) to guide system modeling, with a particular emphasis on Connectivity, Processes, and Interaction Scenario model kinds.

PAACT-BUS incorporates the STRIDE threat identification taxonomy, which categorizes threat impacts on data stores and flows, allowing for adaptable threat modeling at various levels of abstraction.

Unified Architecture Framework (UAF)

UAF is an Object Management Group (OMG) standard that assists in development of architectural descriptions in commercial industry firms, federal government agencies and defense organizations. UAF has a variety of use cases from Enterprise and Mission architecting, to System of Systems (SoS) and Cyber-physical Systems engineering. UAF is based on various modeling languages and frameworks:

- UML, SysML
- DoDAF, MoDAF, NAF

More than just a set of diagrams!

- UAF Modeling Language
- Multiple model kinds
- Multiple viewpoints

STRIDE(S)

STRIDE is a very mature threat hunting method. Invented by Loren Kohnfelder and Praerit Garg in 1999 and adopted by Microsoft in 2002, STRIDE has evolved over time to include new variants STRIDE-per-Element and STRIDE-per-Interaction. The STRIDE mnemonic expands to:

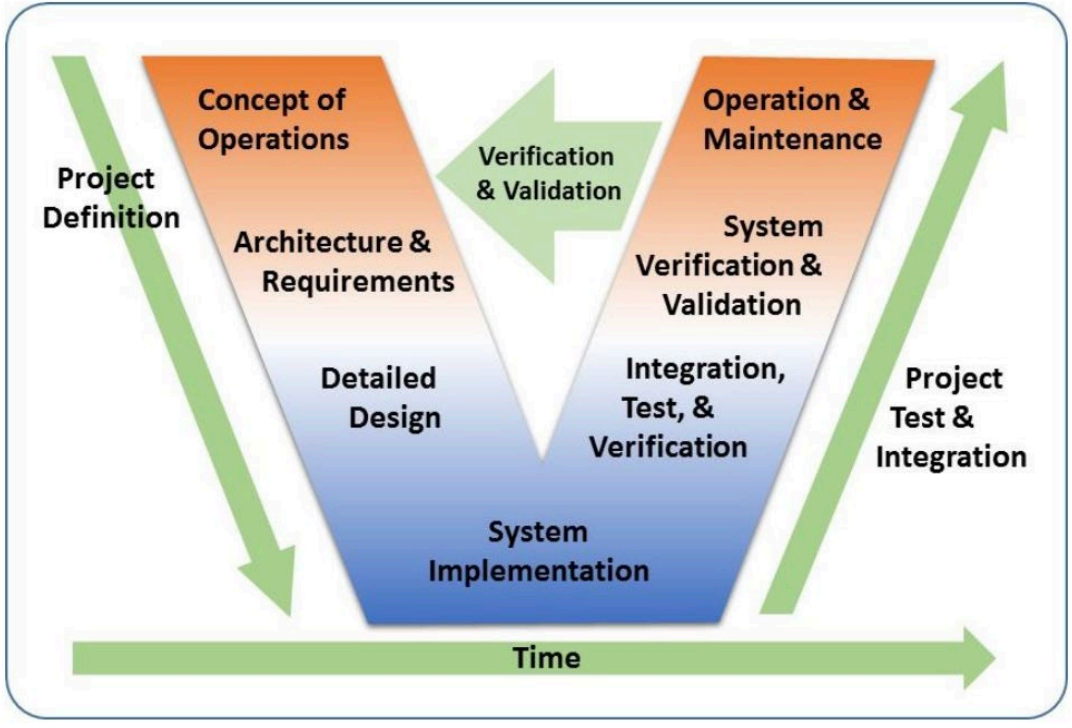
- **S**poofing, **T**ampering, **R**epudiation, **I**nformation Disclosure, **D**enial of Service, **E**levation of privilege

STRIDES extends the STRIDE method with another category of threat: **S**carcity of process.

More than just a mnemonic!
- Guidance on required system design information
- Process for modeling per element or per interaction

Threat Modeling Methods and System Maturity

- STRIDE
- PASTA
- LINDDUN
- CVSS
- Attack Trees
- Persona non Grata
- hTMM
- Quantitative threat Modeling Method
- Trike
- Visual, Agile and Simple Threat
- OCTAVE
- AADL
- STPA-Sec



Graphic from INCOSE [2]

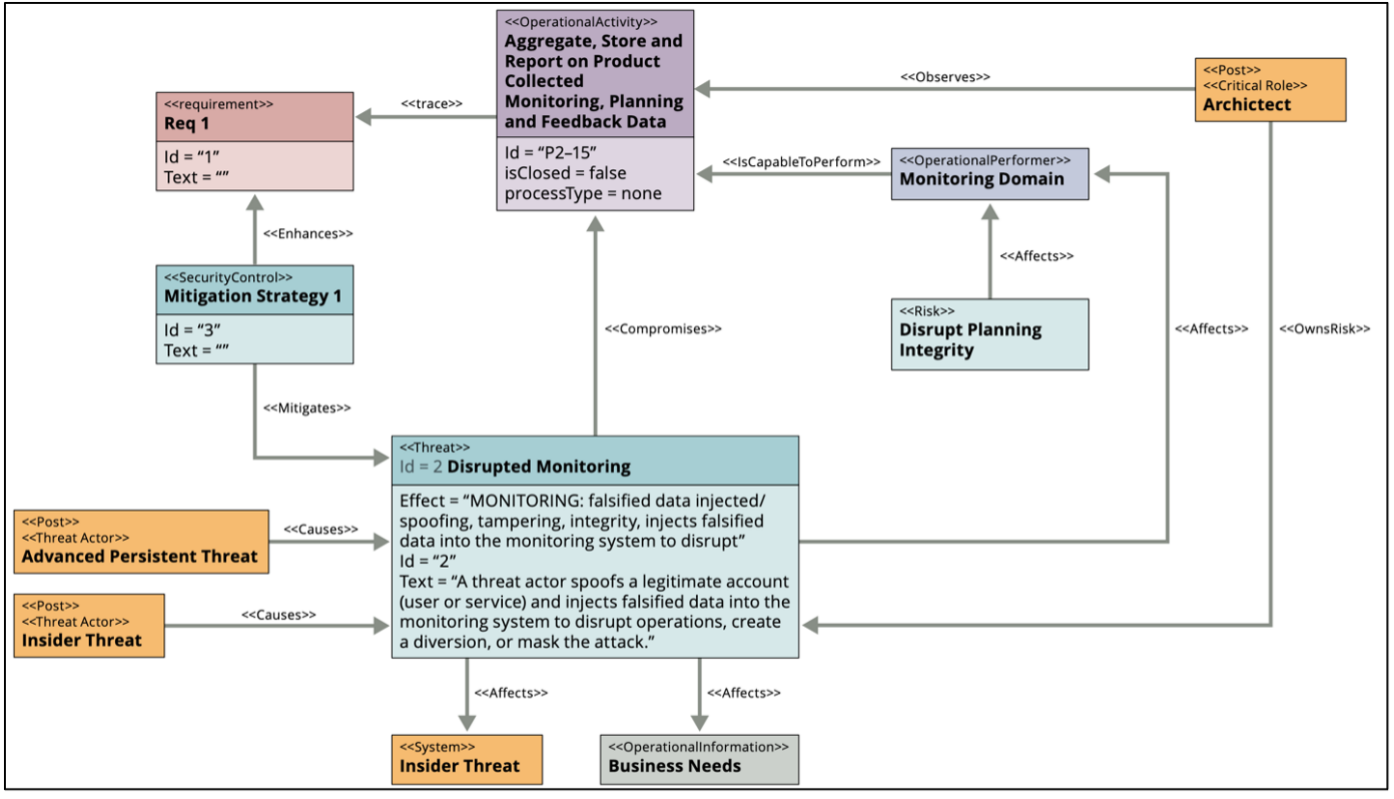
PAACT-BUS Step 1: Model the System

- Determine the scope of the system to be examined
- Prepare the needed UAF model views:
  - Operational Taxonomy (Op-Tx) → Operational Performer structure
  - Operational Connectivity (Op-Cn) → Data and information flows
  - Operational Processes (Op-Pr) → Interactions between performers
  - Personal Taxonomy (Pr-Tx) → Threat actors, insiders

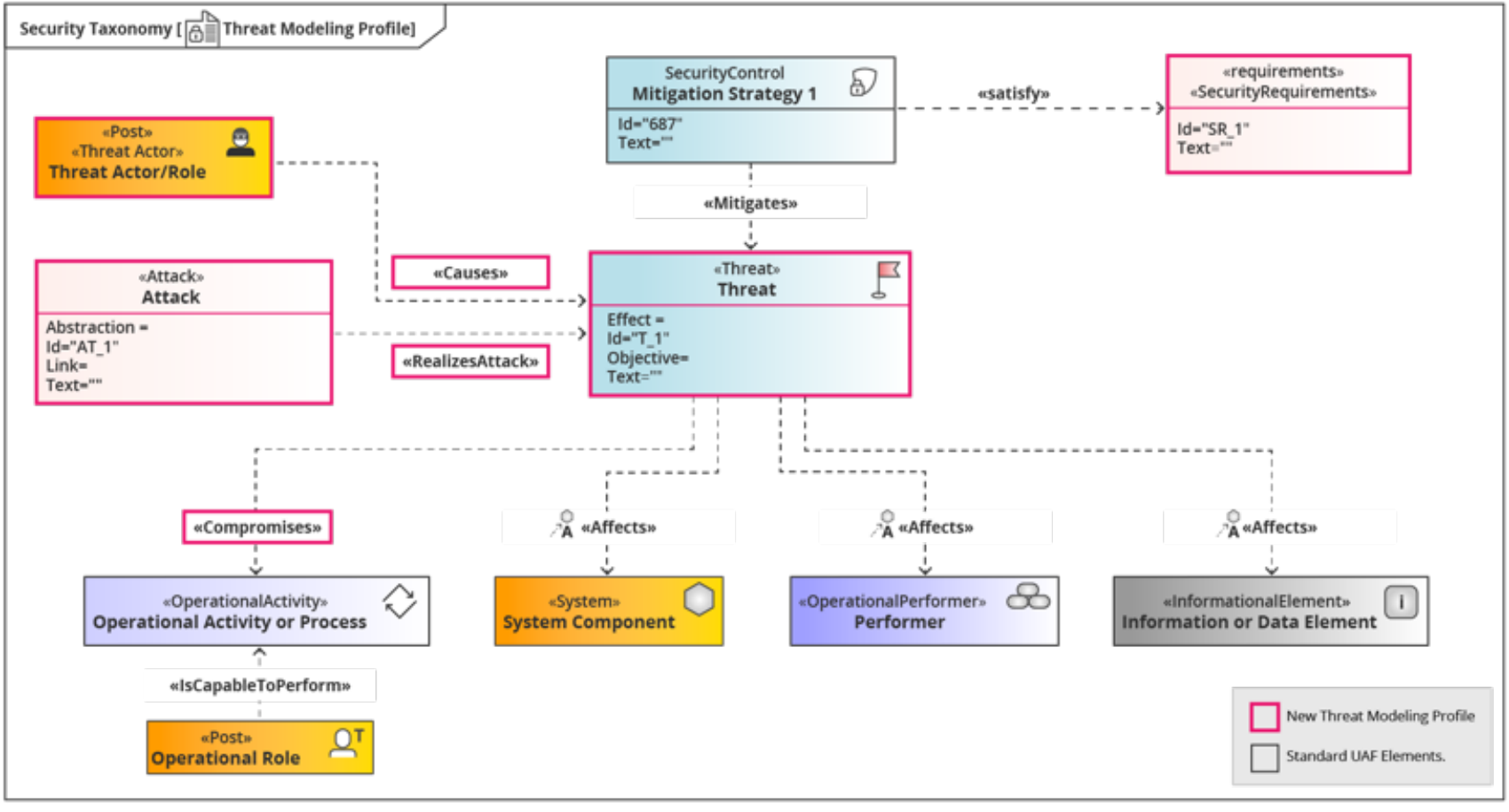
PAACT-BUS Step 2: Apply STRIDES, generate scenarios

- Consider how each threat category could impact the different model elements and modeled interactions
  - Per Element → Operational Taxonomy and Connectivity
  - Per Interaction → Operational Connectivity and Processes
- Capture the threats in a standard format
  - An [ACTOR] performs an [ACTION] to [ATTACK] an [ASSET] to achieve an [EFFECT] and/or [OBJECTIVE].
  - Example: An APT performs a Denial of Service to disrupt the SIEM and insert malware during the outage.

PAACT-BUS Step 3: Generate Sc-Tx Views

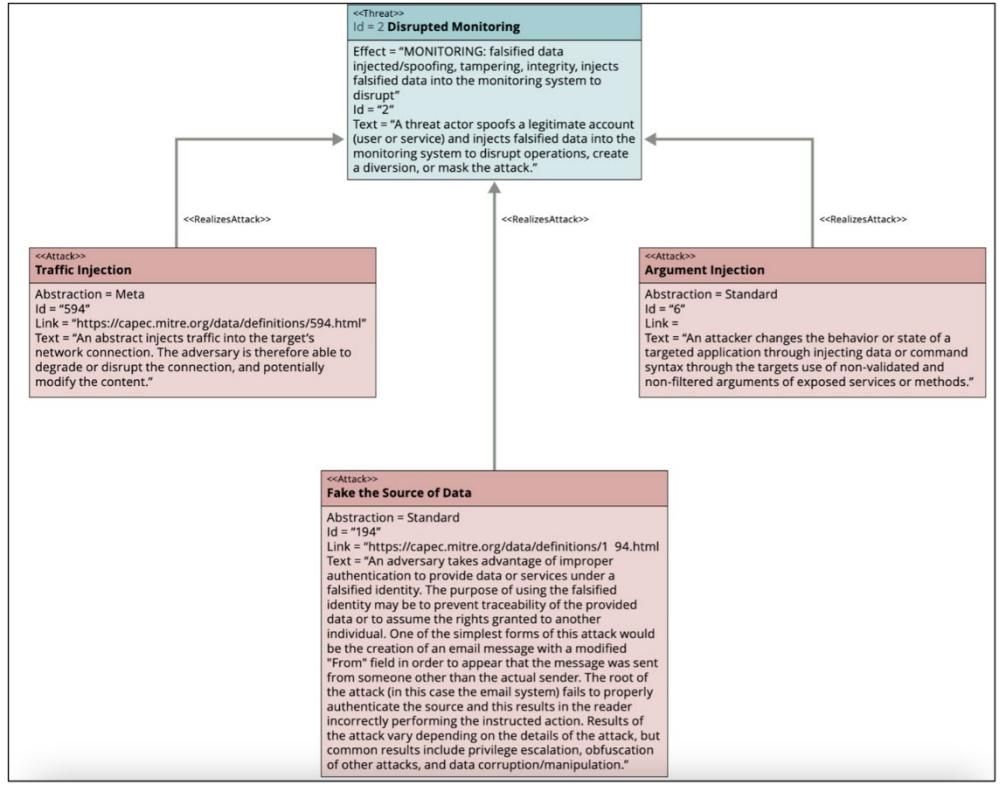


Threat Modeling Profile



PAACT-BUS Step 4: Substantiate the threat model

- Use community driven sources to help identify how a threat actor realizes a given threat scenario and how to mitigate it.
- Example sources of threat TTPs:
  - ATT&CK
  - CAPEC
  - SPARTA
- Examples of security control sources:
  - D3FEND
  - NIST 800-53



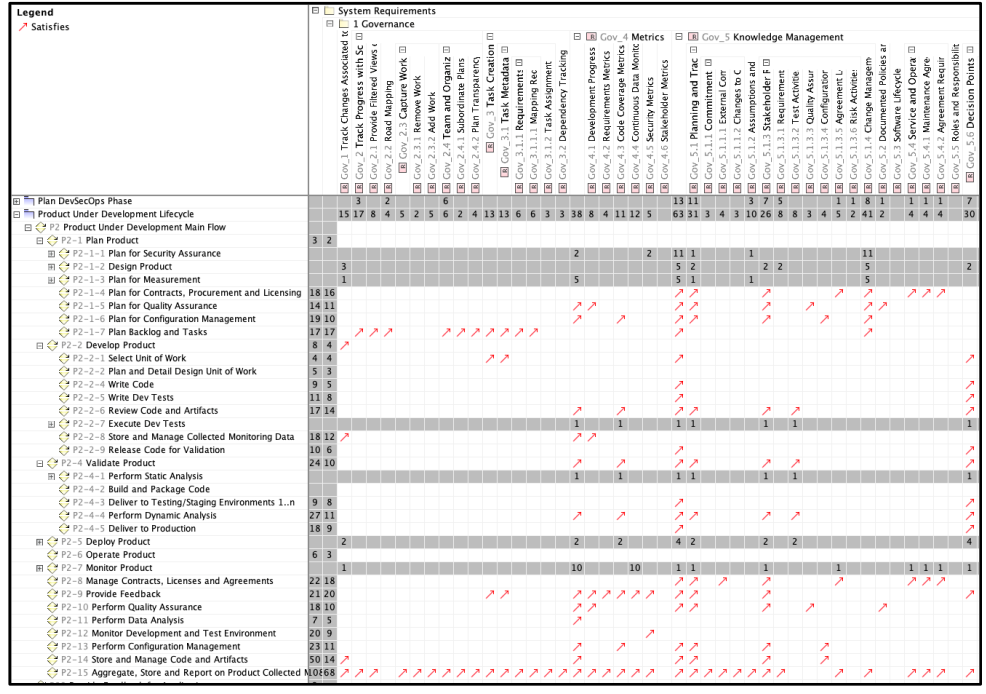
PAACT-BUS Step 5: Analyze the threat model

Use matrices and other model views to identify:

- Key processes, interactions and performers
- Unmitigated threats
- Broadly applicable security controls

Change is constant

- System definition increases
- Architecture changes
- Threats change



Summary

- **MBSE is complex with a steep learning curve but can be a game changer for not only systems engineers but specialty engineering disciplines**
- **Sufficiently large systems require a model to accurately analyze and reason about**
- **Integrating threat modeling with a system model has major benefits**
- **Threat modeling should not be relegated to just a few experts alone with a whiteboard making unsubstantiated claims**

Resources

- https://www.sei.cmu.edu/blog/stop-imagining-threats-start-mitigating-them-a-practical-guide-to-threat-modeling/
- https://www.sei.cmu.edu/library/model-your-way-to-better-cybersecurity/
- DevSecOps PIM: https://cmu-sei.github.io/DevSecOps-Model/#Diagrams__52e7a16f-99c9-418c-b225-b5f934cc58b9

Team Contact

Alex Vesey
Software Engineer

Nataliya Shevchenko
Sr. Engineer

Email: info@sei.cmu.edu