



Analyzing Cyber Security Threats on Cyber-Physical Systems using Model-Based Systems Engineering

Bryan Johnson, Aleksandr Kerzhner, Marc Pomerantz,
Kymie Tan, Brian Campuzano, Kevin Dinkel, Jeremy
Pecharich, Viet Nguyen, and Robert Steele

Jet Propulsion Laboratory California Institute of
Technology, Pasadena, CA 91109

Overview



- Introduction
- Technical Approach
 - Modeling
 - Analysis
 - Visualization
- Issues & Mitigations
- Conclusions



Introduction

- Systems should be resilient to a cyber attack, but increasing frequency, sophistication, and success of adversarial incursions has shown that traditional preventive approaches are insufficient
- Consequently, the toolbox must include an approach that supports resilience. A core need is the ability to perform an impact analysis to support reasoning about the various consequences of adversarial activities.
- Complexity makes impact analysis difficult to accomplish: “siloes”, evolution in the system, and other factors.
- This paper describes a model-based approach aimed at enabling the defense to analyze the various potential consequences of adversarial activities on critical objectives and compose an appropriate response.

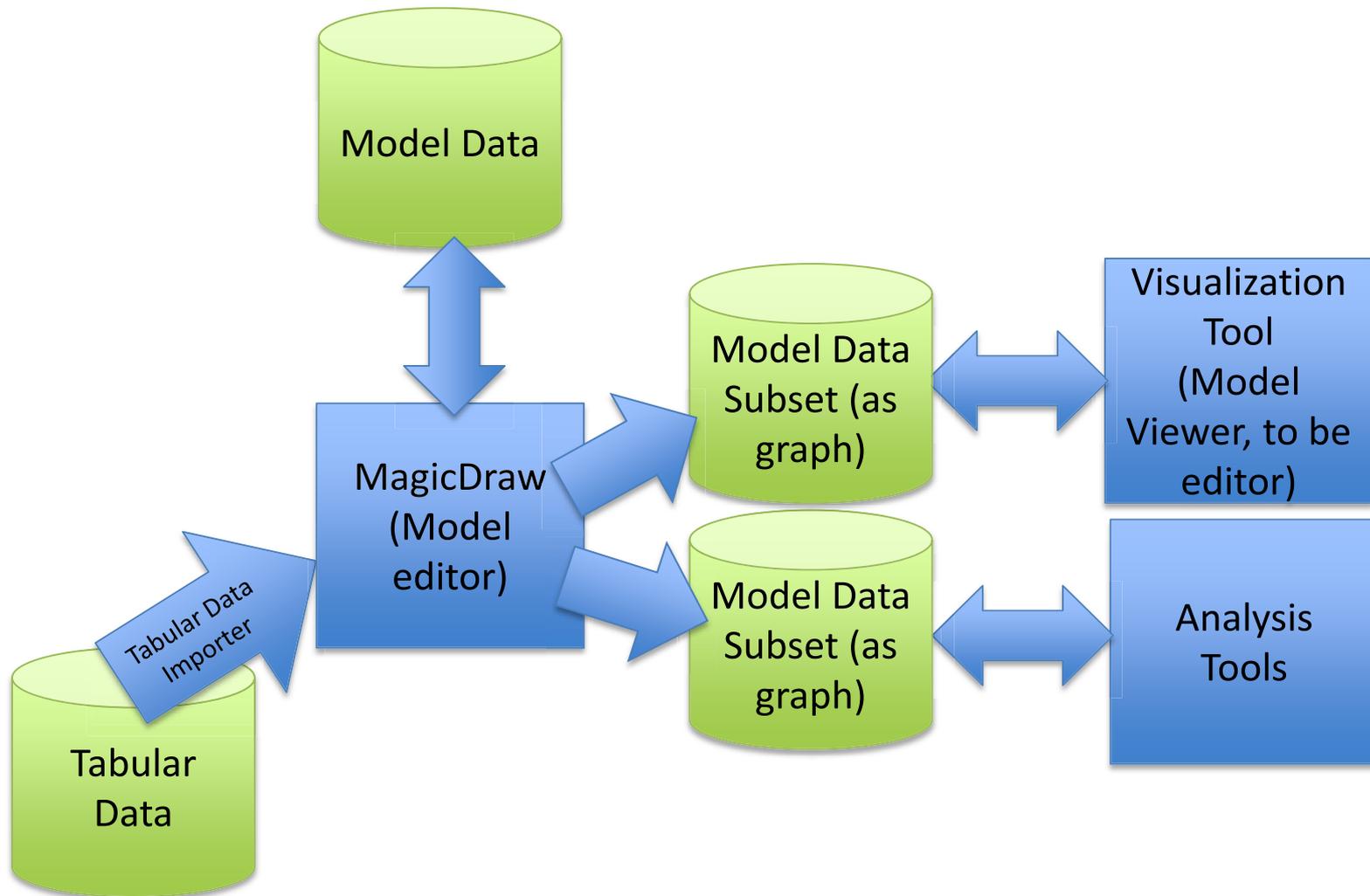
Note: This is ongoing work

Overall Technical Approach

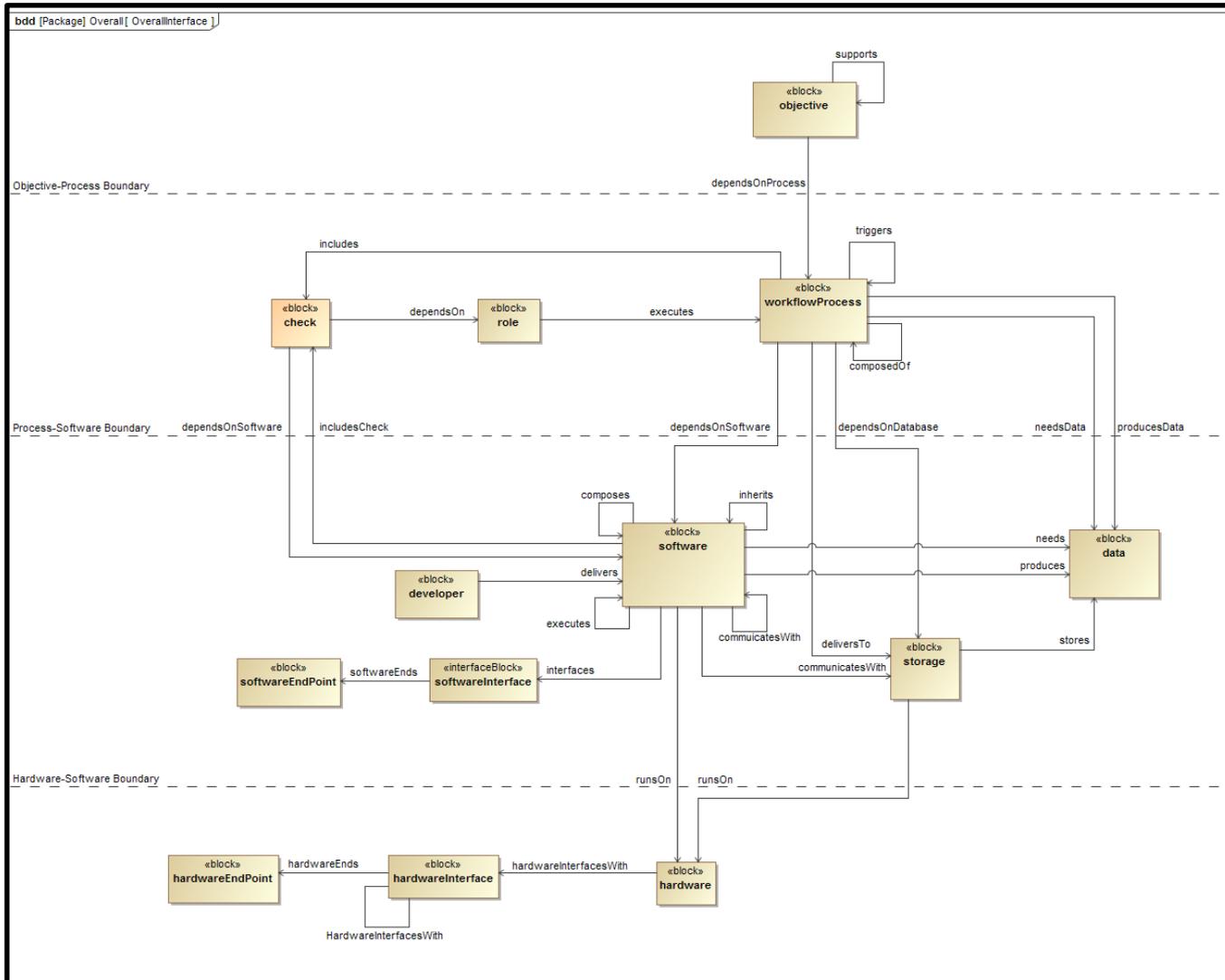


- Goal: Take a “Holistic Approach”
 - Combine disparate information at different levels of abstractions in a common framework to provide a basis for cyber security reasoning
- Capture information about the system:
 - processes, software, data flows, control flows
 - underlying hardware and software infrastructure (operating systems, protocols, ...) – include CVEs (common vulnerabilities and exposures)
- Capture interconnections between abstraction levels (cyber attacks “move through system”)
- Develop visualize tool where SMEs can inspect system, search for assets, and navigate abstraction levels
- Identify risks by analyzing modeled system:
 - What-if attack propagation
 - Potential paths from end points to system assets, processes, or mission objectives
 - Centrality/interconnection/resilience of system assets
- Verify model using SMEs interactions and information about real system

Knowledge Capture Approach



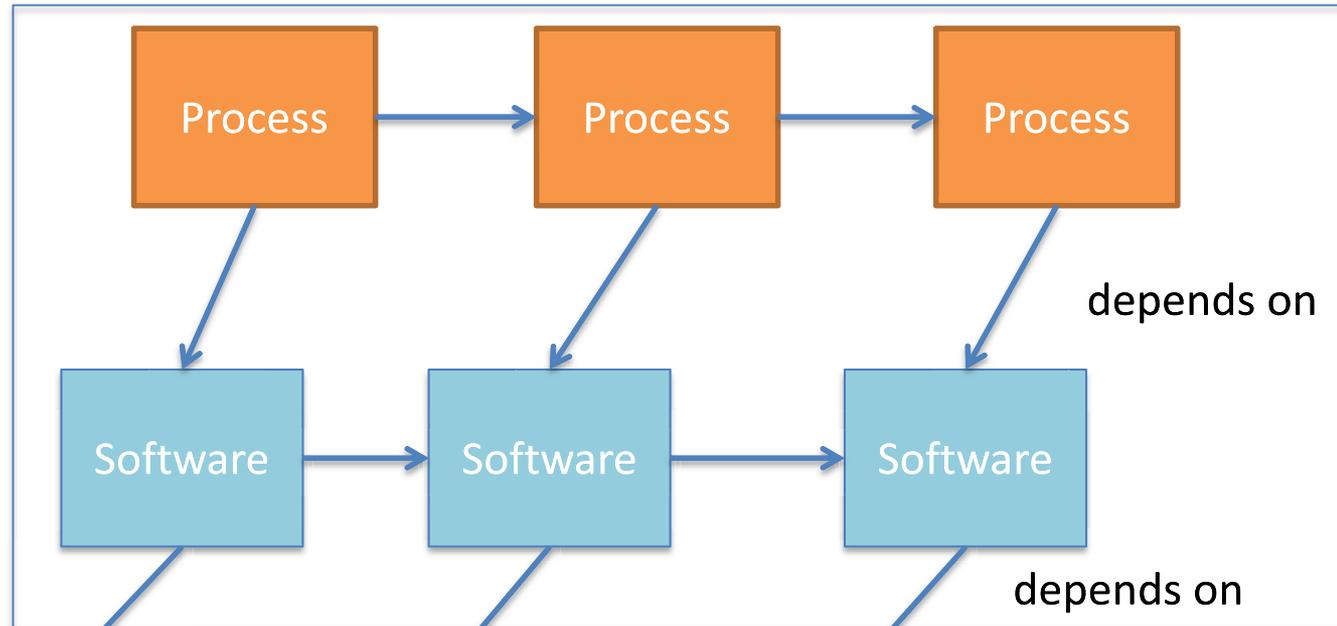
(Simplified) Model Architecture



Analysis Approach

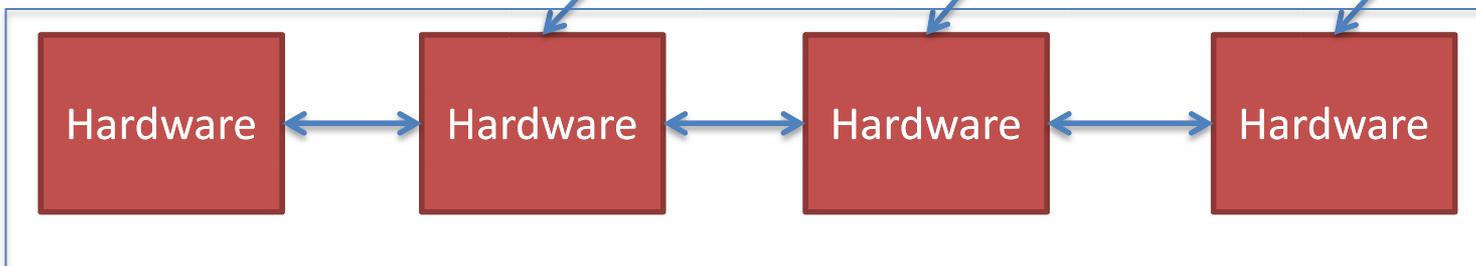


Related to
Business
Processes



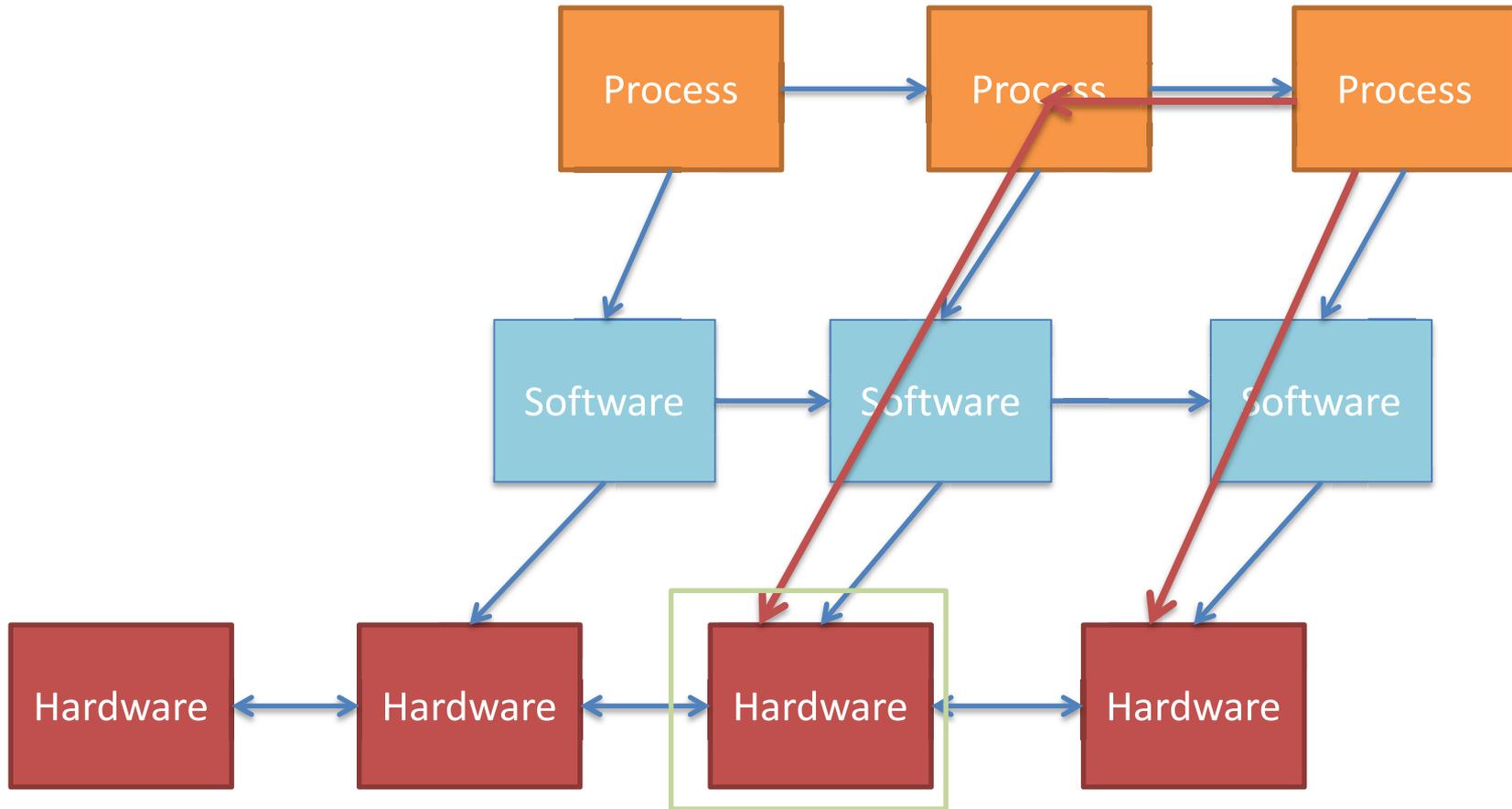
depends on

depends on



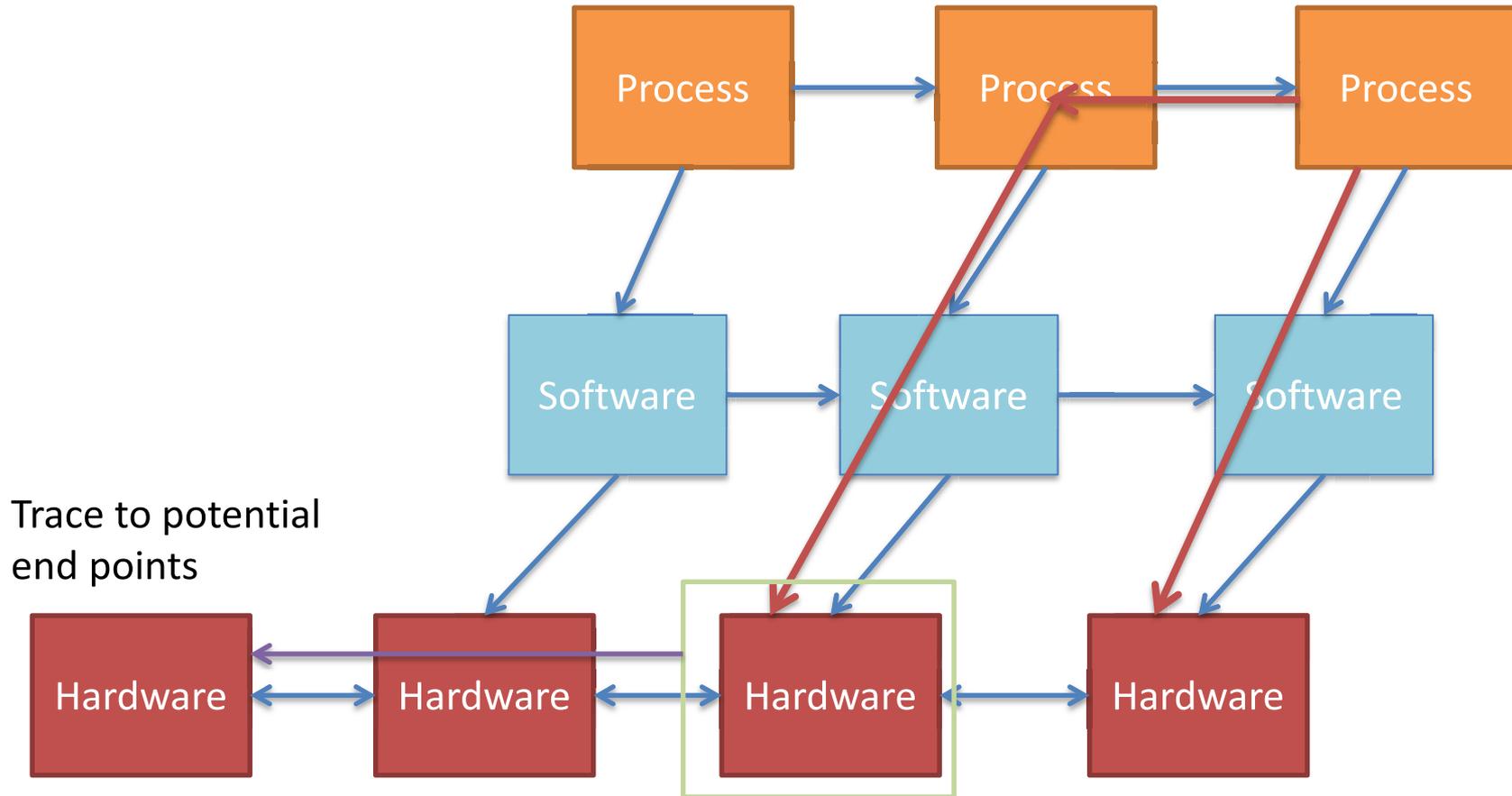
Underlying
infrastructure,
Hardware, OS
“Library”
Software,
Network, etc.

Analysis Approach



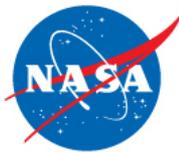
Identify potential
underlying
vulnerabilities

Analysis Approach

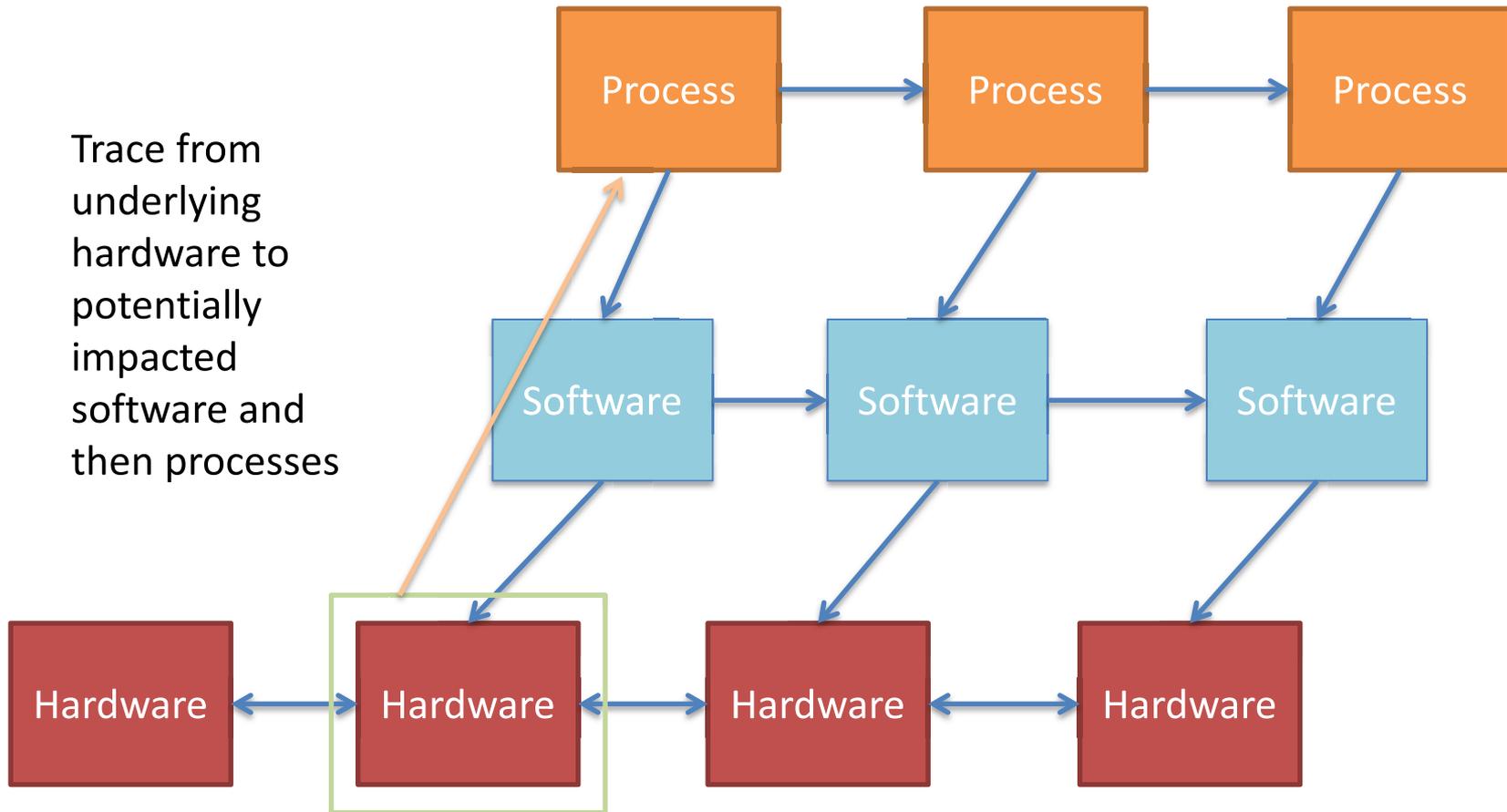


Trace to potential end points

Identify potential underlying vulnerabilities



Analysis Approach



Trace from underlying hardware to potentially impacted software and then processes

Identify potential underlying vulnerabilities

Visualization System



CCARAT

Select All Select None Zoom In Zoom Out Vertex Labels Paint Selection Lasso Selection Rectangle Selection Hide Selection Show Selection Isolate Selection Clear Isolation

Select Neighbors Select Parents Select Children Select Incident Edges Select Incoming Edges Select Outgoing Edges Select Edge Vertices Select Edge Sources Select Edge Targets

Entity List

Edges Vertices Visible Only

Search Results

Quick search name...

Index	TypeName	SimpleType
0	V	Software
1	V	Software
2	V	Software
3	V	Software
4	V	Software
5	V	Software
6	V	Software
7	V	Software
8	V	Software
9	V	Software
10	V	Software
11	V	Software
12	V	Software
13	V	Software
14	V	Software
15	V	Software
16	V	Software
17	V	Software
18	V	Software
19	V	Software
20	V	Software
21	V	Software

Entity Properties

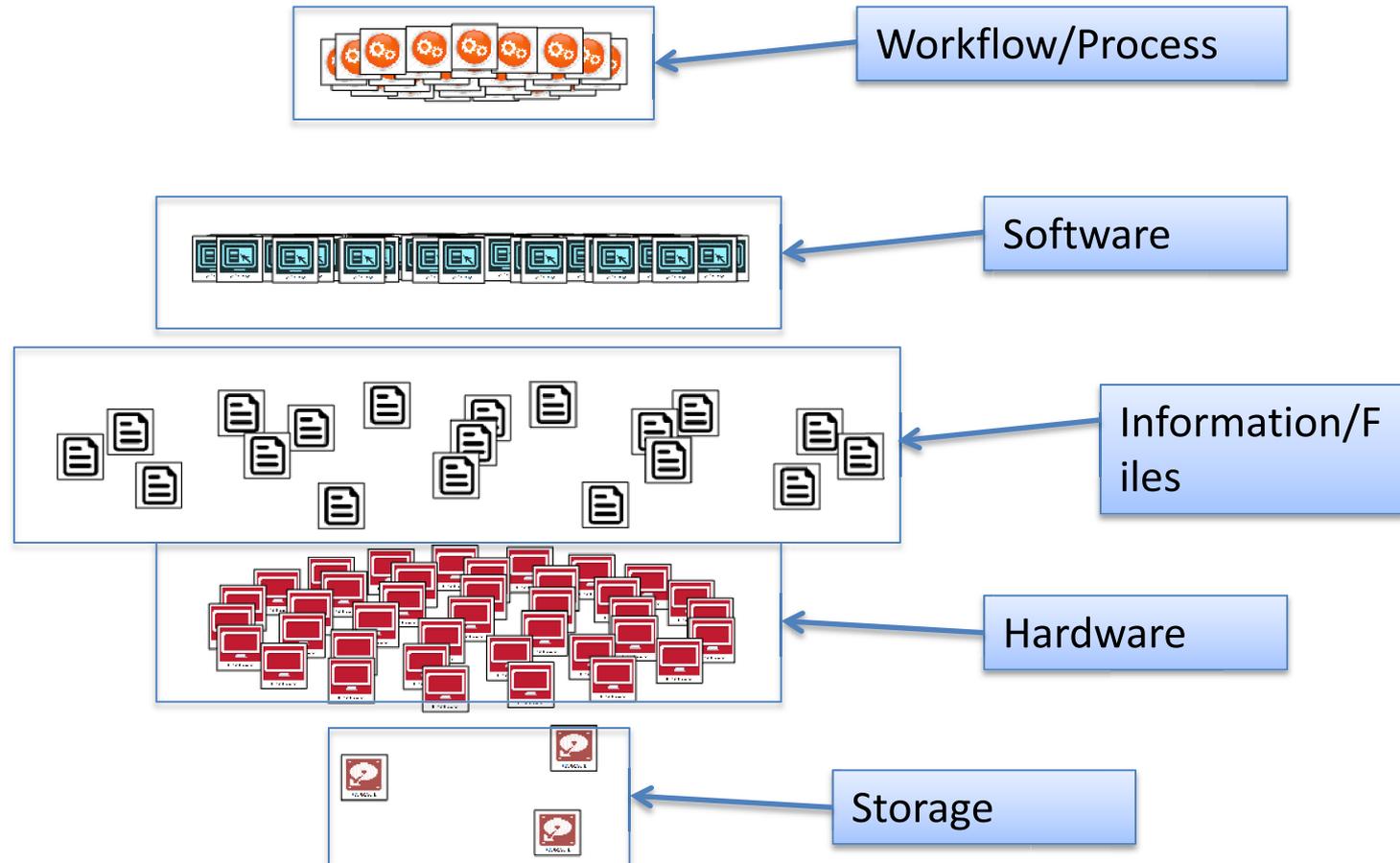
Automatically Expand

Index	TypeKey	Value
-------	---------	-------

Analysis Toolbox

- Find Adjacent Vertices**
Finds all vertices adjacent to the currently selected vertices.
[Run](#)
- Central Nodes**
Finds the five nodes with the highest degree.
[Run](#)
- Depends On**
Finds the hardware elements that the currently selected elements depends on.
[Run](#)
- Processes Impacted**
Finds all Process Elements impacted by the selected elements.
[Run](#)
- Ping Status**
Pings all selected vertices, treating the name of the vertex as a server name or IP address.
[Run](#)
- Point Failure**
Selects system's most connected vertex, also selects adjacent vertices.
[Run](#)
- Weighted Shortest Path**
Finds one of the most vulnerable paths between two vertices.
[Run](#)

Model of System – Different Levels



Visualizing Analysis Results



The screenshot displays a network analysis application with the following components:

- Entity List:** A table listing hardware entities with columns for Index, Type Name, and SimpleType. The selected row is:

Index	Type Name	SimpleType
155	V	Hardware
158	V	Hardware
157	V	Hardware
158	V	Hardware

- Entity Properties:** A table showing properties for the selected entity, including OS, Open_Source, center, displayName, failed_vertex, ping_successful, and simpleType.

Index	TypeKey	Value
V	OS	Open Source
V	Open_Source	center
V	displayName	failed_vertex
V	failed_vertex	ping_successful
V	ping_successful	simpleType

- Search:** A search results table showing the attribute 'ping_successful' with a value of 'ne None'.

Attribute	Op	Value
ping_successful	ne	None

- Analysis Toolbox:** A collection of analysis tools with 'Run' buttons:
- Find Adjacent Vertices:** Finds all vertices adjacent to the currently selected vertices.
- Central Nodes:** Finds the five nodes with the highest degree.
- Depends On:** Finds the hardware elements that the currently selected elements depend on.
- Processes Impacted:** Finds all Process Elements impacted by the selected elements.
- Ping Status:** Pings all selected vertices, treating the name of the vertex as a server name or IP address.
- Point Failure:** Selects system's most connected vertex, also selects adjacent vertices.

Current State and Advances



Current State	Advances
Lack of Cyber and Mission SME access to captured information	Simplify access and manipulation by developing visualization tool
Cyber SME's unable to navigate information in real time, e.g., adjust focus	Intuitive interface for real-time exploration of the entire model
Traditional models lack critical detail, e.g., cyber information	Model now includes protocols, underlying libraries, versions, CVEs as part of metadata
Unable to rapidly update model information to keep pace with cyber environment	Confirm validity of model w.r.t. the real system, scan/query systems in real time.
Unable to propagate change through the model – attack propagation	Analyses that propagate attack vectors, visually displayed
Veracity of evidence is not clear	Capture sources in the model with dates to identify when information is valid



Conclusions and Next Steps

- By holistically considering system, can perform what-if analyses and understand impact on process
- Developing a reusable strategy for visualization and analysis that could be applied to multiple problems.
- Current Progress:
 - Developed initial model of the system
 - Identified analysis types and implemented initial analyses
 - Developed initial visualization tool prototype to iterate with SMEs
- Future Work:
 - Automated attack tree exploration
 - Incorporate timing information
 - Continue to develop visualization look, feel, and performance

REFERENCES



Related Works

- **Model Construction** [Burgess2004, BM2011]
 - Graph-based model
 - Servers, Firewall, Routers, Databases, Software, Files, Workflow Processes
 - Edges represent a relationship between different components
- **Incorporating NVD** [AY2008,FSWJ2008, FW2008]
 - Use NVD to focus on privilege escalation
- **Mission Impacts** [Goodall2009,Grimaila2009,Thiem2005]
 - Map files to mission objectives
 - Identify critical mission components
- **Cyber Environment** [Jakobson2011,Ralston2007]
 - Military
 - SCADA
 - Space-based asset



Bibliography

- [AY2008] B. Argauer and S. Yang. VTAC: virtual terrain assisted impact assessment for cyber attacks. *Proc. SPIE 6973, Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security*, 2008.
- [Burgess2004] M. Burgess et al. A graph theoretical model of computer security. *International Journal of Information Theory*. 3(2) 70-85, 2004.
- [Bau2011] J. Bau and J. Mitchell. Security modeling and analysis. *Security and Privacy*. (9)3, 18-25, 2011.
- [Frigault2008a] M. Frigault, et al. Measuring network security using dynamic Bayesian network. *Proceedings of ACM workshop on Quality of protection*, 2008.
- [Frigault2008b] M. Frigault and L. Wang. Measuring network security using Bayesian network-based attack graphs. *Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International*, 2008.
- [Goodall2009] J. Goodall et al. Camus: Automatically mapping cyber assets to missions and users. *Military Communications Conference, 2009*.
- [Grimaila2008] M. Grimaila et al. Improving the cyber incident mission impact assessment (CIMIA) process. *Proceedings of the 4th annual workshop on Cyber security and information intelligence research*, 2008.



Bibliography

[Jakobson2011] G. Jakobson. Mission cyber security situation assessment using impact dependency graphs. *Proceedings of the 14th International Conference on Information Fusion (FUSION)*, 2011.

[Liu2005] Y. Liu and H. Man. Networking vulnerability assessment using Bayesian networks. *Proc. SPIE 5812, Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security*, 2005.

[Musman2010] S. Musman et al. Evaluating the impact of cyber attacks on missions. *Proceedings of the 5th International Conference on Information Warfare and Security*, 2010.

[Ralston2007] P.A.S Ralston et al. Cyber security risk assessment for SCADA and DCS networks. *ISA Transactions*. (46)4, 583-594, 2007.

[Thiem2005] L. Thiem. A study to determine the damage assessment methods or models on Air Force networks. Air Force Inst of Tech Wright-Patterson AFB OH School of Engineering and Management, 2005.