# Mission Aware MBSE Meta-Model

# Mission Aware (MA) Meta-Model Overview

SysML v2 is proposed standardization target for the formalization of associations between Systems Theoretic Process Analysis (STPA), Model-Based System Engineering (MBSE), and Mission Aware (MA) concepts.

**STPA**

STPA Handbook
Leveson & Thomas - 2018

**MBSE**

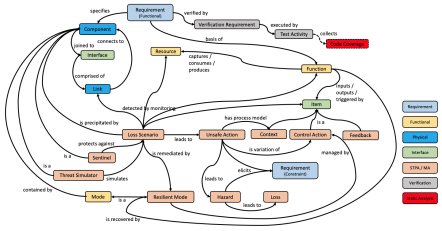
OMG
SysML v2 RFP - 2017

**Mission Aware**

SERC
2012-2019

**MA MBSE Meta-Model**

MISSION AWARE



# STPA Overview



STPA is an iterative, methodical hazard analysis technique to identify causes of hazardous conditions intended to improve or promote system safety.

- In cyber-physical systems, security can be treated as analogous to safety.

## STPA Outputs and Traceability

*Figure 2.21* shows the traceability that is maintained between various STPA outputs.

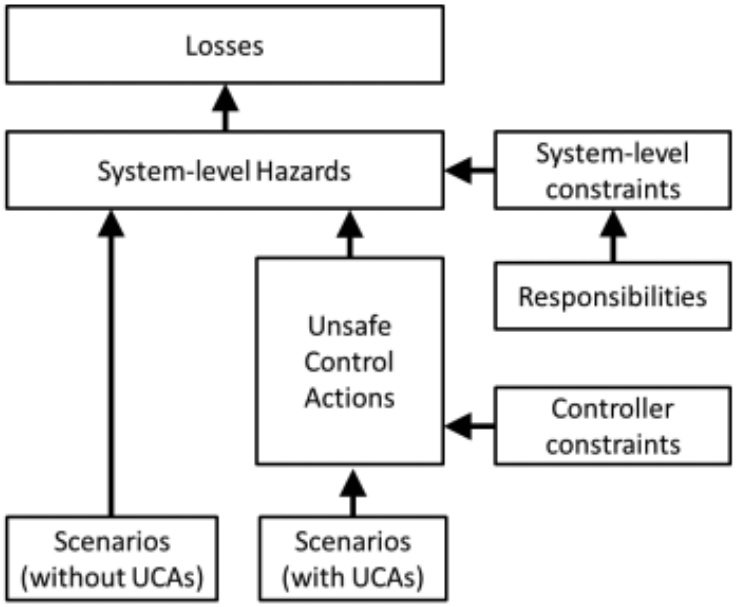
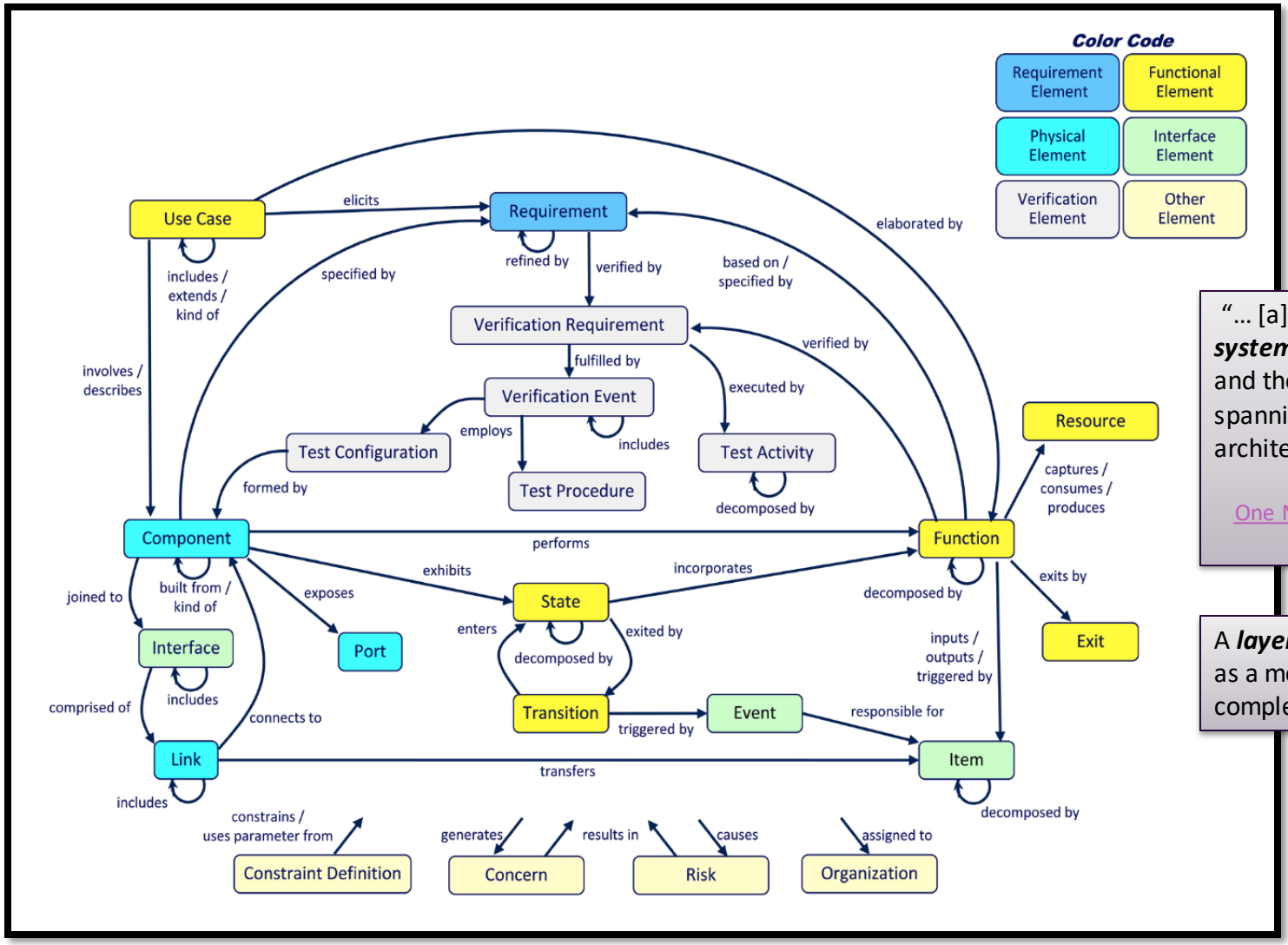



*Figure 2.21: Traceability between STPA outputs*

- A ***Loss*** involves something of value to stakeholders. Losses may include a loss of human life or human injury, property damage, environmental pollution, loss of mission, loss of reputation, loss or leak of sensitive information, or any other loss that is unacceptable to the stakeholders.
- A ***Hazard*** is a system state or set of conditions that, together with a particular set of worst-case environmental conditions, will lead to a loss.
- An ***Unsafe Control Action*** (UCA) is a control action that, in a particular context and worst-case environment, will lead to a hazard.
- A ***Loss Scenario*** describes the causal factors that can lead to the unsafe control and to hazards.

# MBSE Meta-Model Overview

Key requirement defined by Object Management Group (OMG) for SysML v2 is *"a meta-model of core SE concepts with precise semantics."* Vitech Corporation MBSE meta-model largely aligns with SysML v2 goals.




"… [a] representation of critical ***systems engineering concepts*** and their ***interrelationships*** spanning requirements, behavior, architecture, and test."

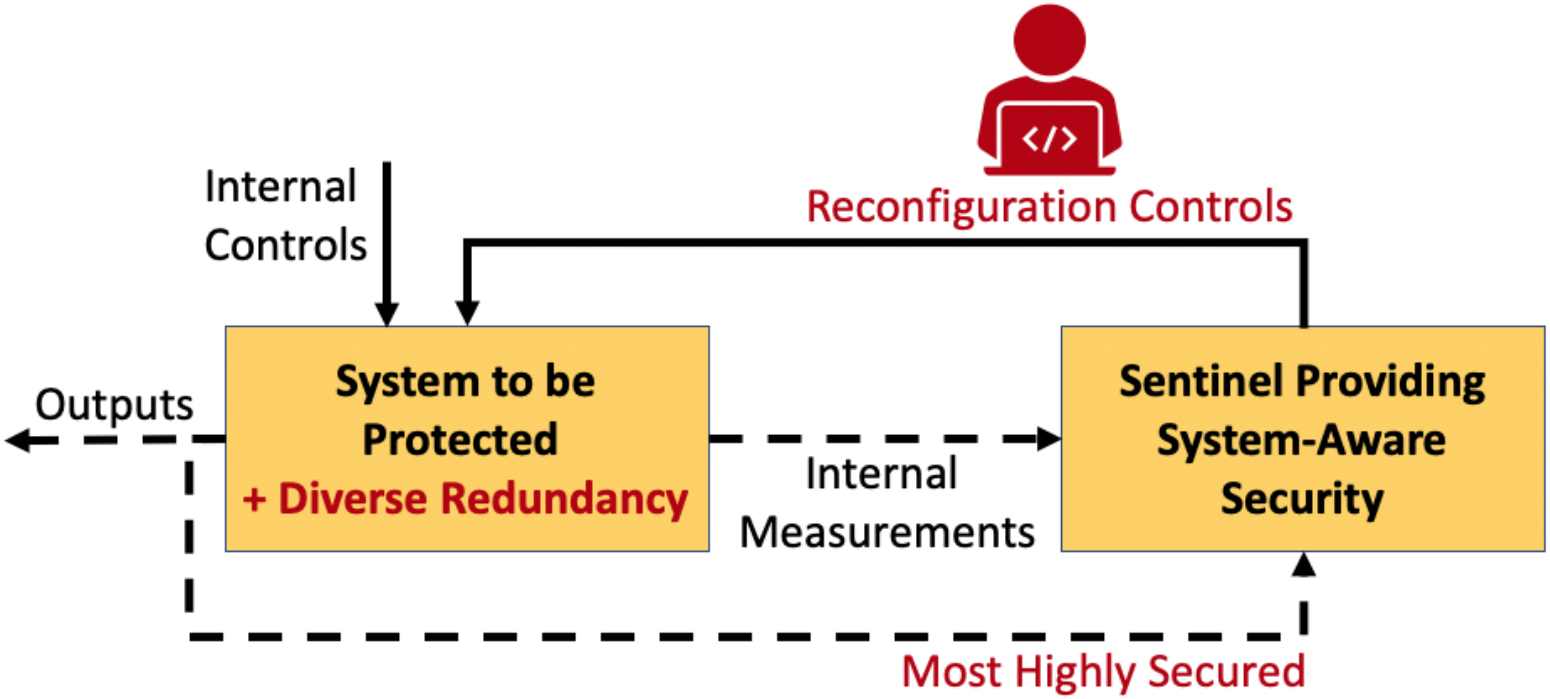
One Model, Many Interests, Many Views - Vitech 2018

A ***layered / hierarchical*** model as a mechanism to manage complexity.

# Mission Aware Overview

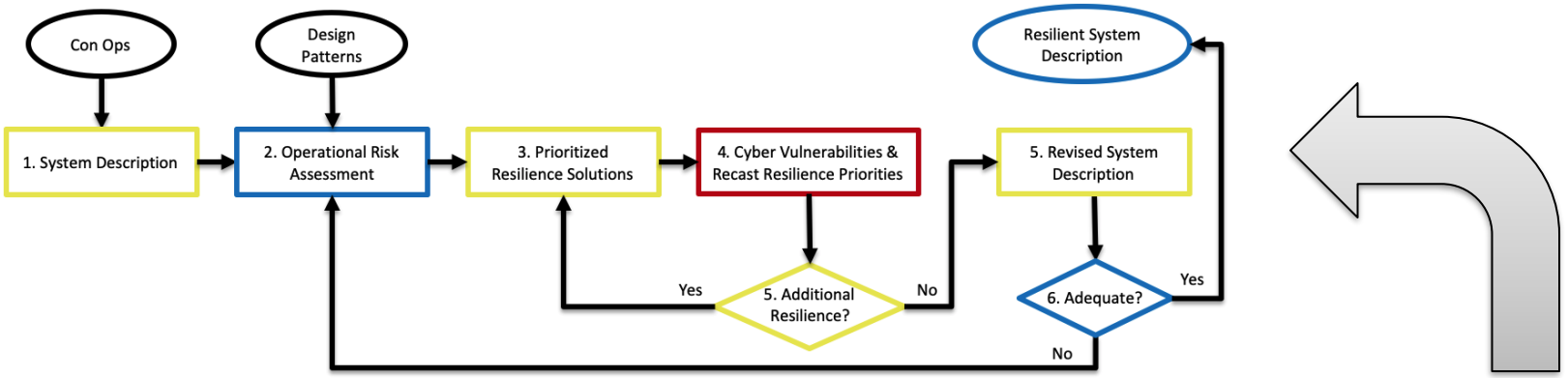


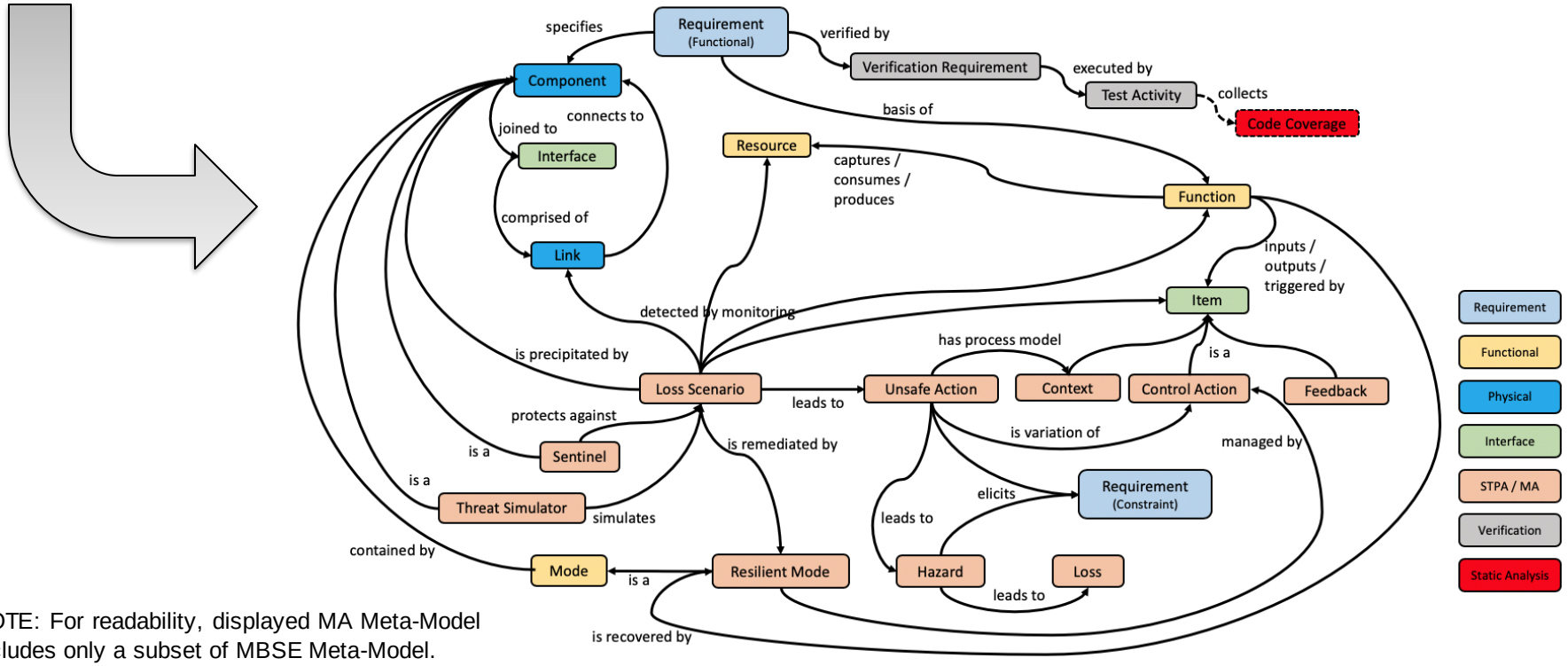
- A ***Resilient Mode*** is a distinct and separate method of operation of a component, device, or system based upon diverse redundancy. Resilience allows the system to maintain a safe level of operational normalcy in response to anomalies, including threats of malicious and unexpected nature.
- A ***Sentinel*** is responsible for monitoring and reconfiguration of a system using available Resilient Modes. The Sentinel subsystem is expected to be far more secure than the system being addressed for resiliency.



# CSRM / MA Meta-Model Mapping

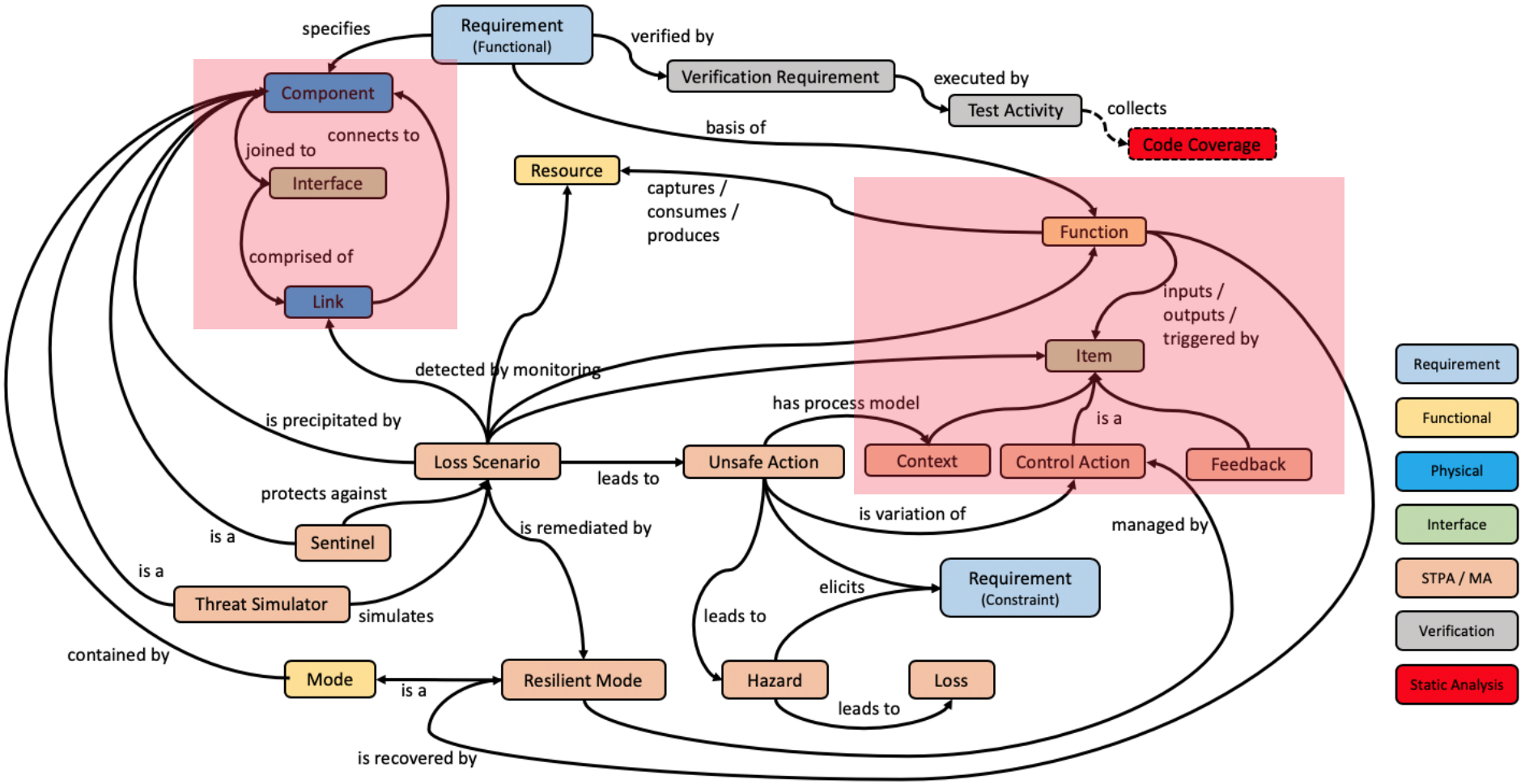




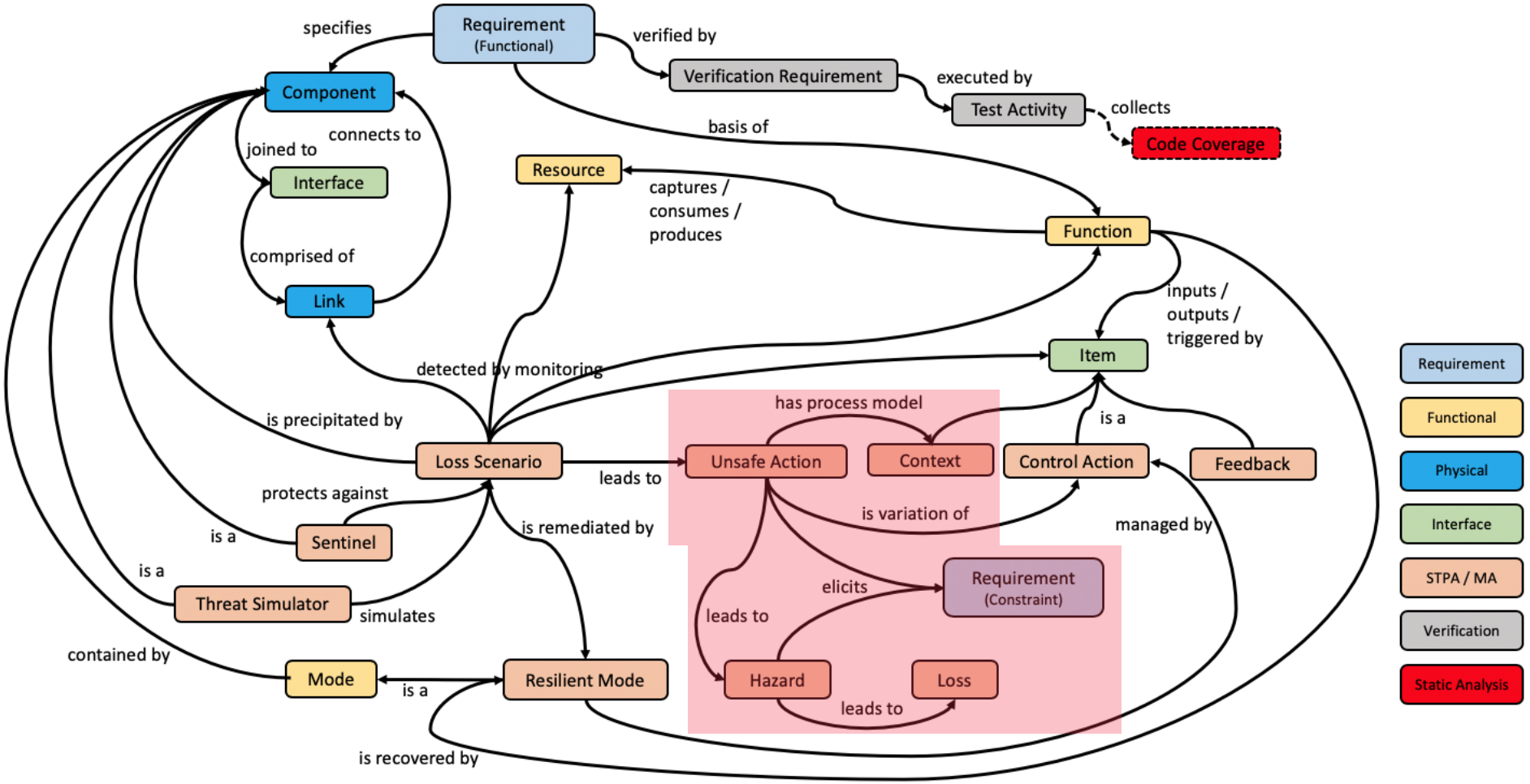

NOTE: For readability, displayed MA Meta-Model includes only a subset of MBSE Meta-Model.

# CSRM Step #1 – System Description



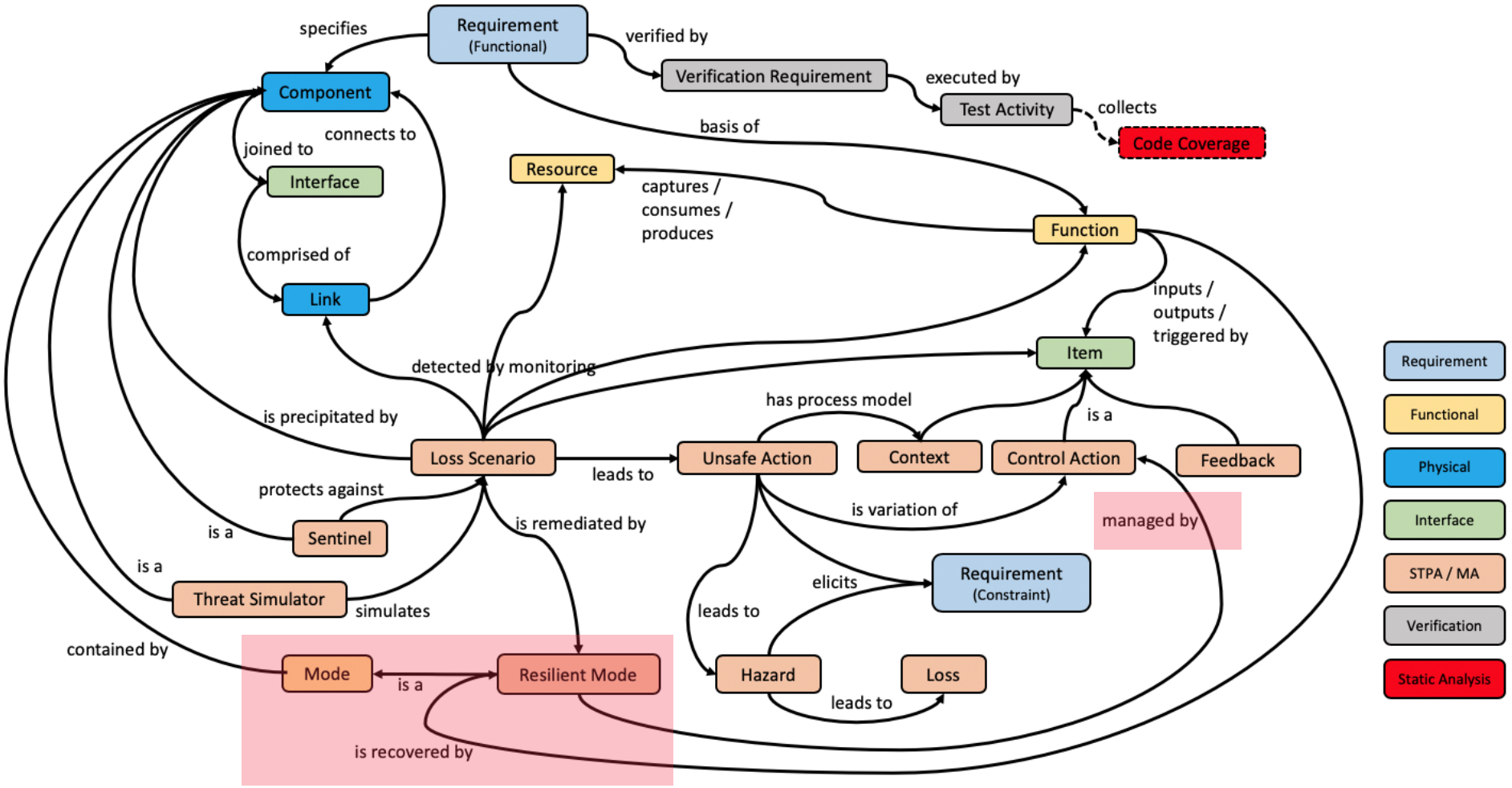
# CSRM Step #2: Operational Risk Assessment



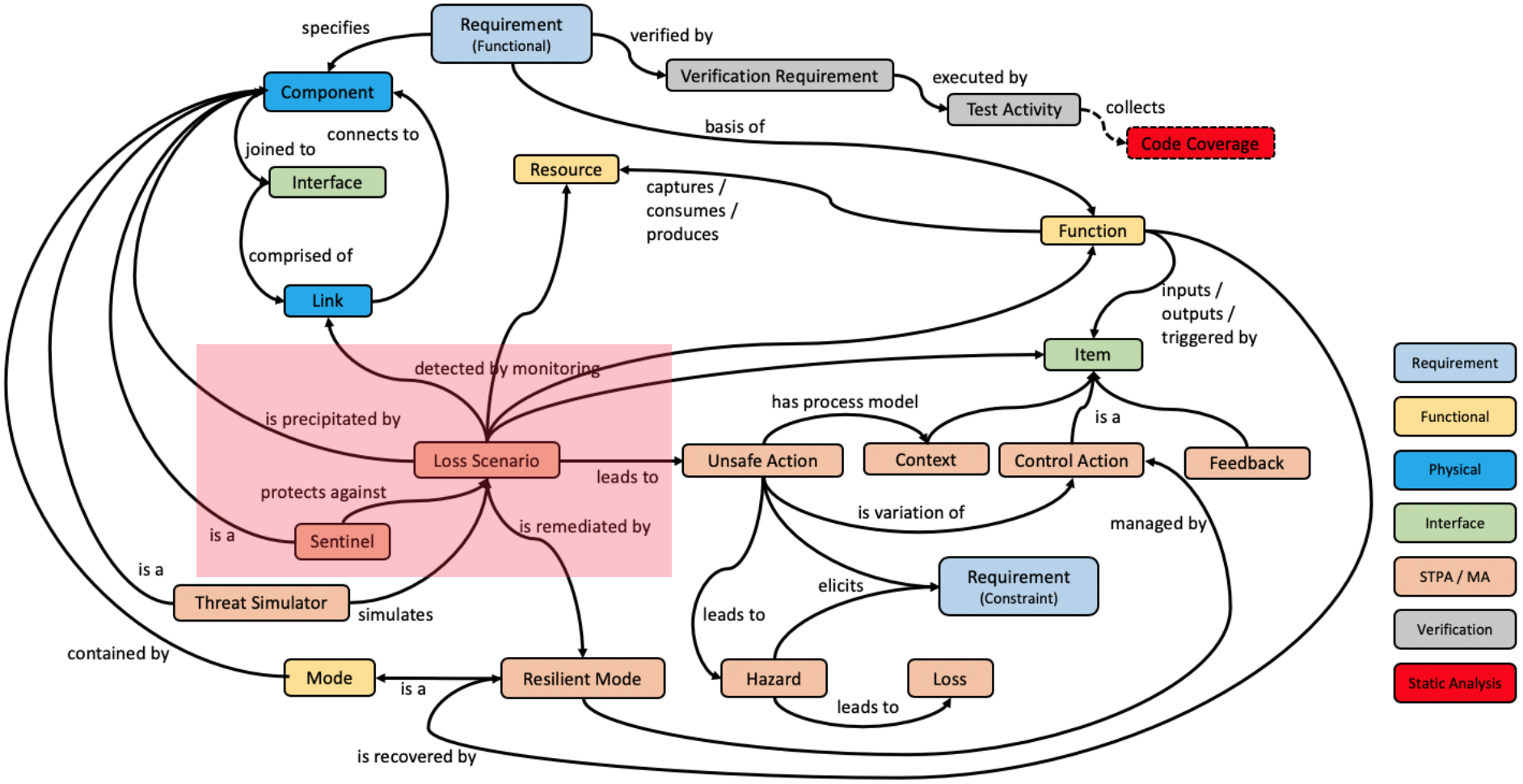


# CSRM Step #3: Prioritized Resilient Solutions





# CSRM Step #4: Cyber Vulnerabilities & Recast Resilient Priorities



# CSRM Step #5/6: Resilience Evaluation



## ***Mission Aware***: MBSE Attributes and Metrics

| Object | Attribute | Values | Notes |
|---|---|---|---|
| Loss | missionImpact | High / Med / Low | Blue Team |
| Loss Scenario | attackLikelihood | High / Med / Low | Red Team |
| | attackType | External<br>Insider<br>SupplyChain | |
| | attackPattern | <CAPEC-#>:<Title> | |
| | detectionPattern | DataConsistency<br>ChangingControlInput<br>Introspection | |
| | detectionTime | seconds | Time budget to detect loss |
| | isolateTime | seconds | Time budget to isolate loss via system /component tests. |
| Resilient Mode | complexity | High / Med / Low | Number of model "contained by" associations. Indication of cost. |
| | effectiveness | High / Med / Low | Impact on remediating High "likelihood" attacks associated with High "mission impact". |
| | operationalImpact | High / Med / Low | Degree of operator training need.<br>Degree of mission interruption. |
| | restoreTime | seconds | Time budget to restore system function via resilient mode. |
| | operatorDecisionTime | seconds | Time budget for operator decision time to enable resilient mode.<br>0 implies automated resilient mode. |

# CSRM Step #5/6: Resilience Evaluation



***Recovery Ratio***: A mechanism to evaluate & refine a System Architecture against defined Resiliency requirements:

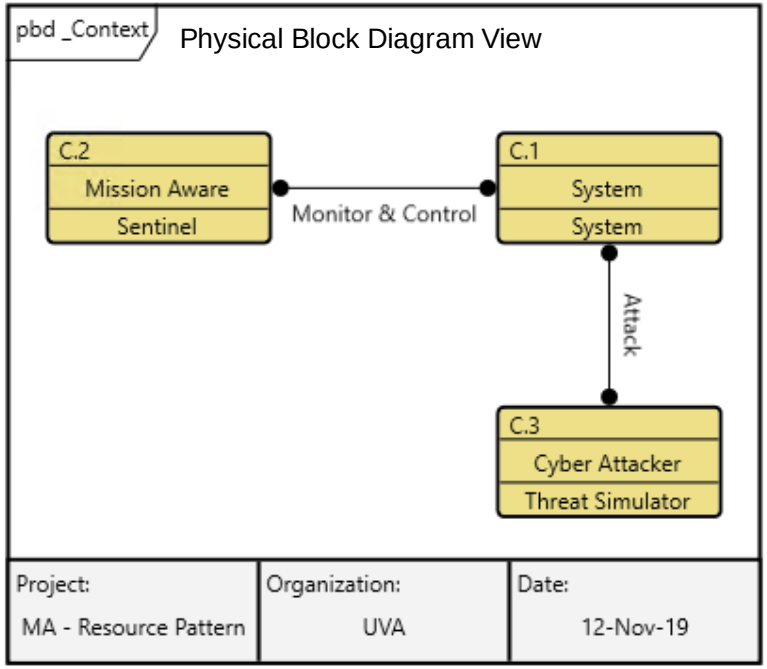
- An iterative process as system design is refined / matured

| Metric | Units | System Model Evaluation / Simulation |
|---|---|---|
| Resilient Mode:<br>"Recovery Ratio" per System Function [per Loss Scenario]<br><br>*Calculated:* Measured / Expected | < 1: Acceptable<br>> 1: Not Acceptable | Recovery time includes:<br>• Detection<br>• Isolation<br>• Restoration<br>Including:<br>• Technical: System Components<br>• Operational: System-of-System Interactions<br>• Operator: Expected Decision Times |
| Loss Scenario: Time to Detect | seconds / minutes | Impact tradeoff for Sentinel interfaces:<br>• polling-based (system / link loading)<br>• event-based, etc. |
| Loss Scenario: Time to Isolate | seconds / minutes | Impact tradeoff for System / Component Test capabilities |
| Resilient Mode: Time to Restore | seconds / minutes | Impact tradeoff for Resilient Modes:<br>• Active/Active<br>• Active/Standby (Hot / Warm / Cold)<br><br>Includes Operator decision time |

# Example: Behavior Model Simulation

# Example: Architecture Model






Loss Scenario – Attack Pattern:

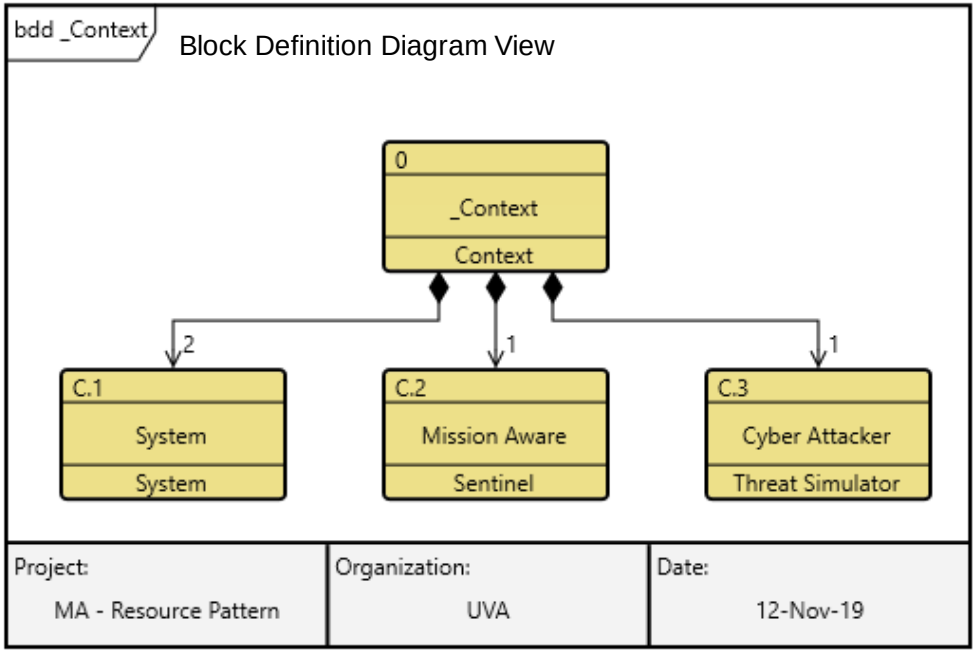
- CPU Overload
- CAPEC-443: Malicious Logic Inserted Into Product Software by Authorized Developer

Sentinel - Design Pattern:

- Resource Introspection - CPU Idle Time

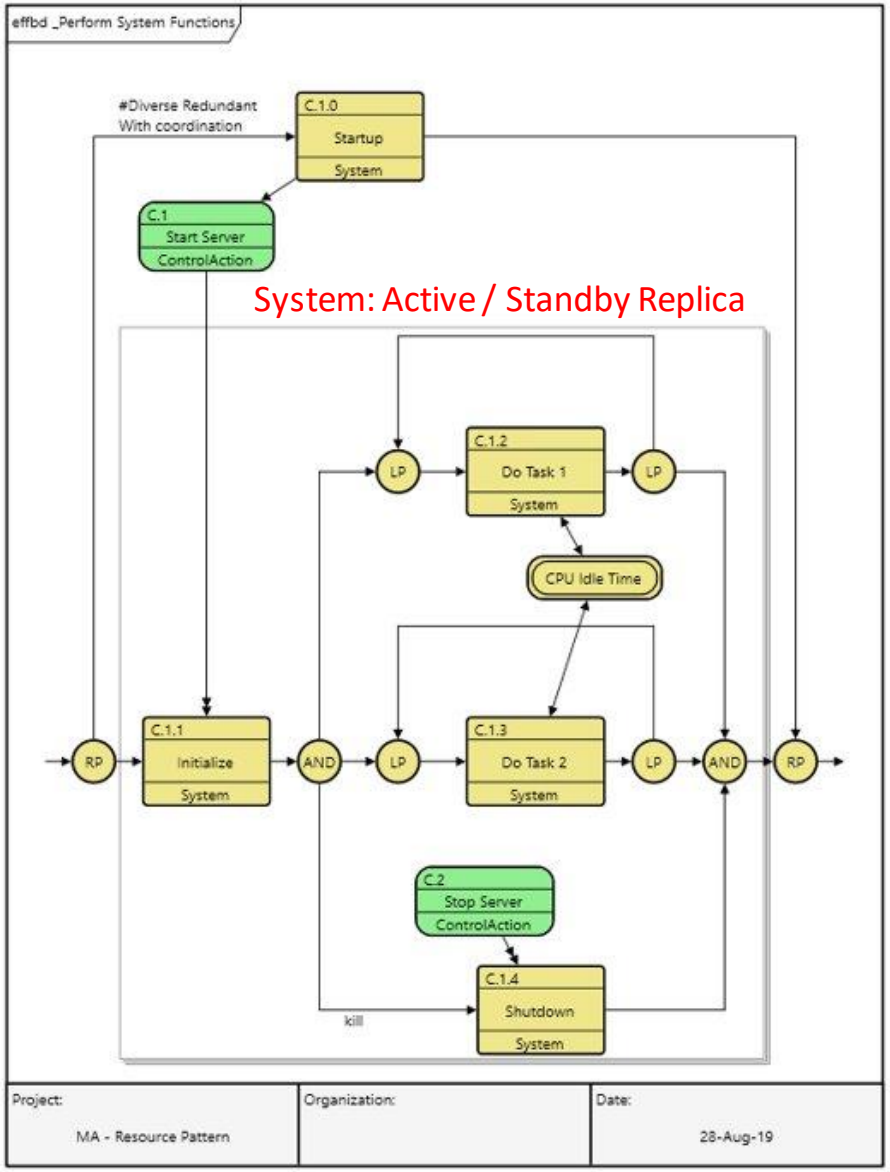
Resilient Mode:

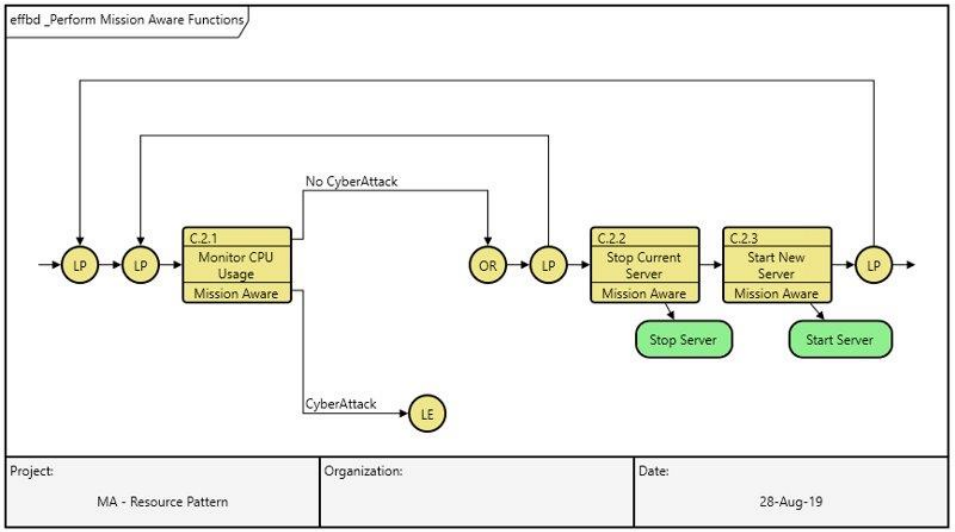
- Active / Standby

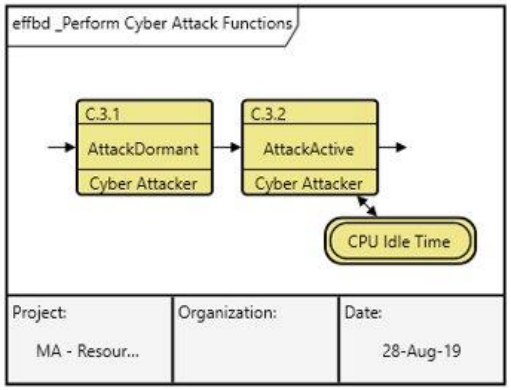


# Example: Behavior Model






The Enhanced Functional Flow Block Diagram (EFFBD), like its SysML cousin the activity diagram, is a complete representation of behavior. EFFBDs unambiguously represent the *flow of control* through sequencing of functions as well an overlay of *data* and *resource* interactions.

# Example: Simulation Transcript