Resilience Engineering

April 1, 2021

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Resiliency

Definition: The capability to recover quickly from difficulties; toughness

Goal: Instead of reacting to events, design and implement a DevSecOps system that supports application development and deployment that identifies, monitors, and ultimately adjusts to events both expected and unexpected.

Approach:

(1) Identify the metrics of resiliency and how they relate to resilient operation of both a system supporting DevSecOps and the development and deployment of software in such a system.

(2) Collaborating with other ongoing research at the SEI including the development of a Platform Independent Model (PIM), TwinOps, AADL, ML model development, and using Cornerstone to create an exemplar prototype of a resilient DevSecOps system.

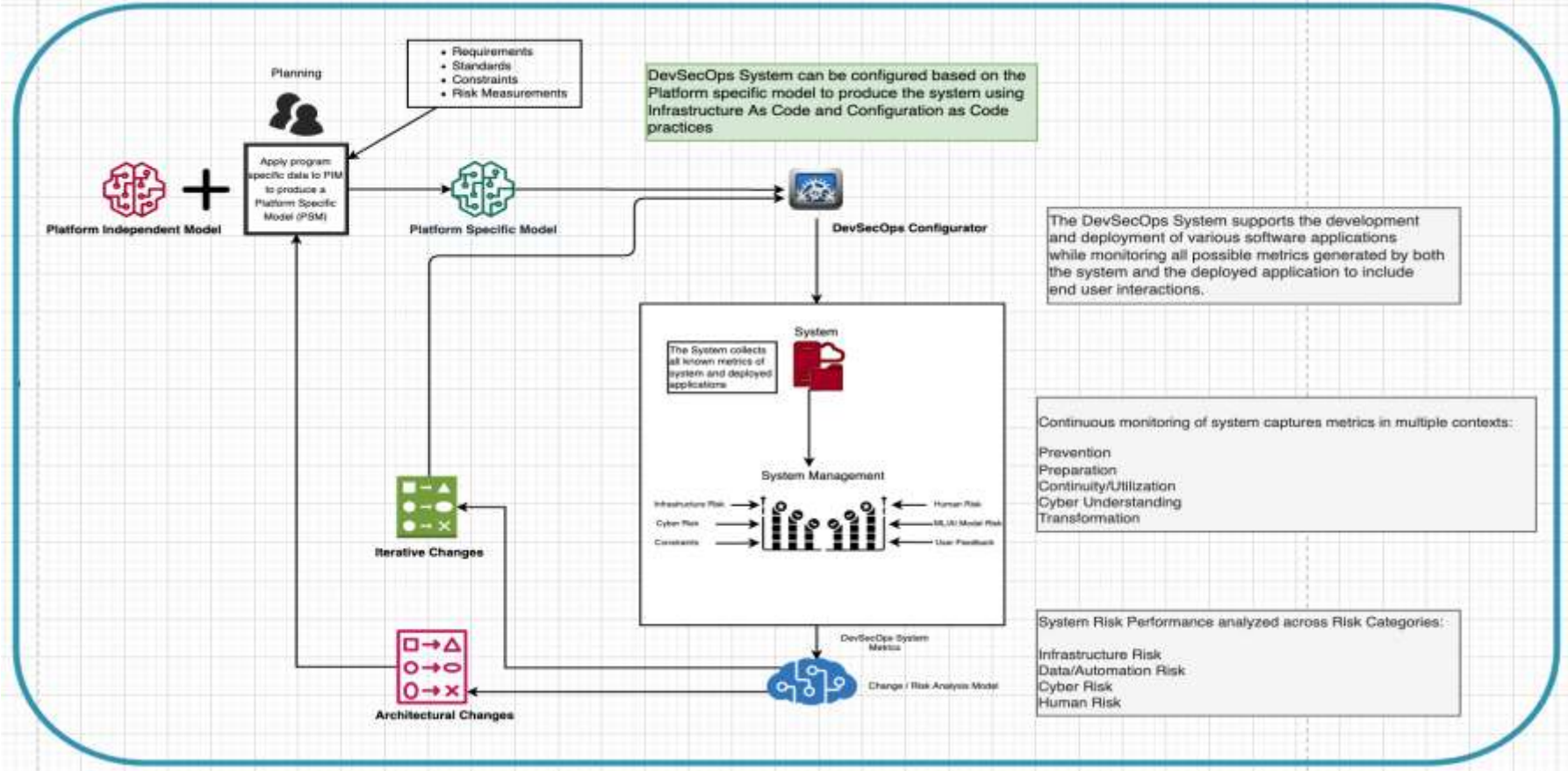
This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM21-0313

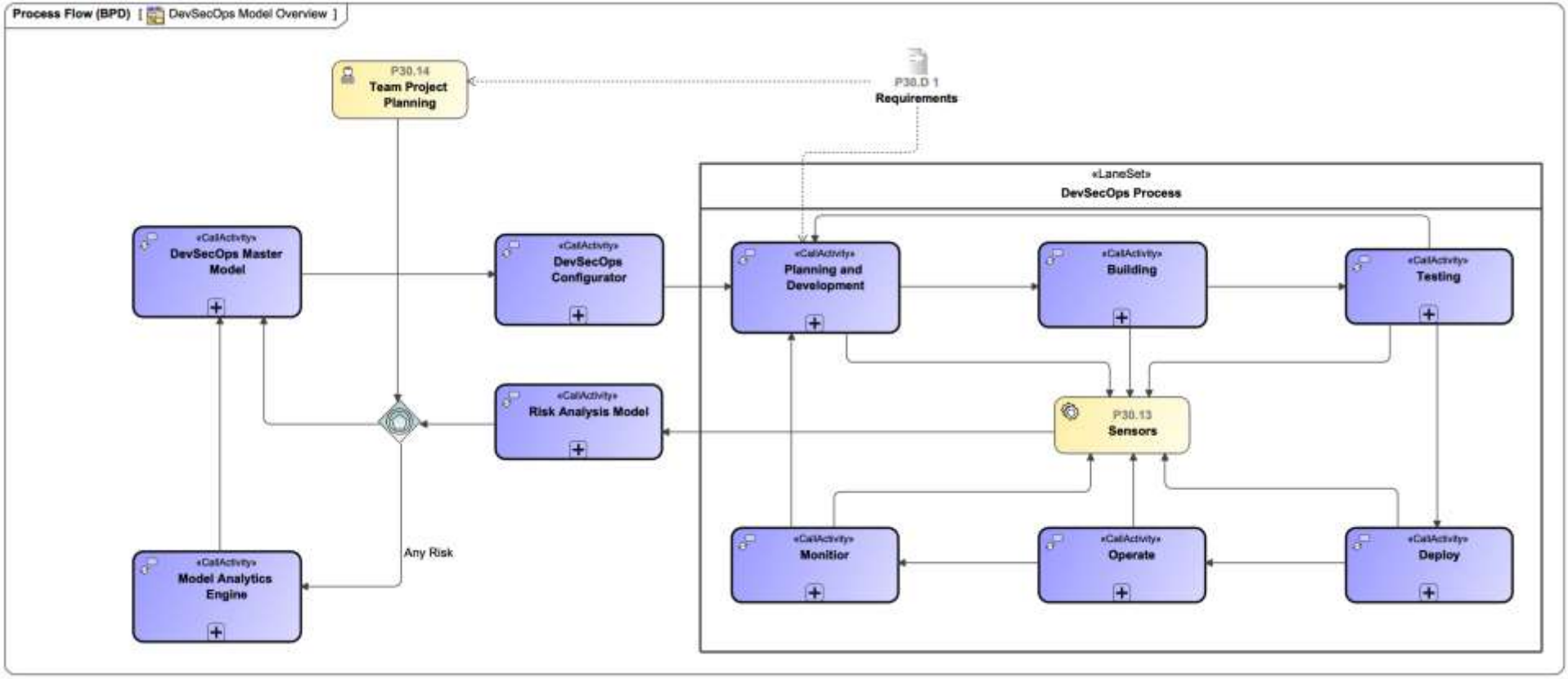
Approach

- Containerization utilized throughout system
- Enumerate all cyber resiliency metrics, identify most relevant to DevSecOps pipeline
- **Create Resiliency Use cases to determine relevant high priority metrics for DevSecOps**
- Identify data sources that contribute to high priority DevSecOps resiliency metrics
- Identify the metrics for a risk analysis model, and subsequent Action(s) that can be taken
- Test and validate data sources with actual system and application (Cornerstone)
- Build a MBSE model for risk analysis
- Incorporate data sources into MBSE to model resiliency response

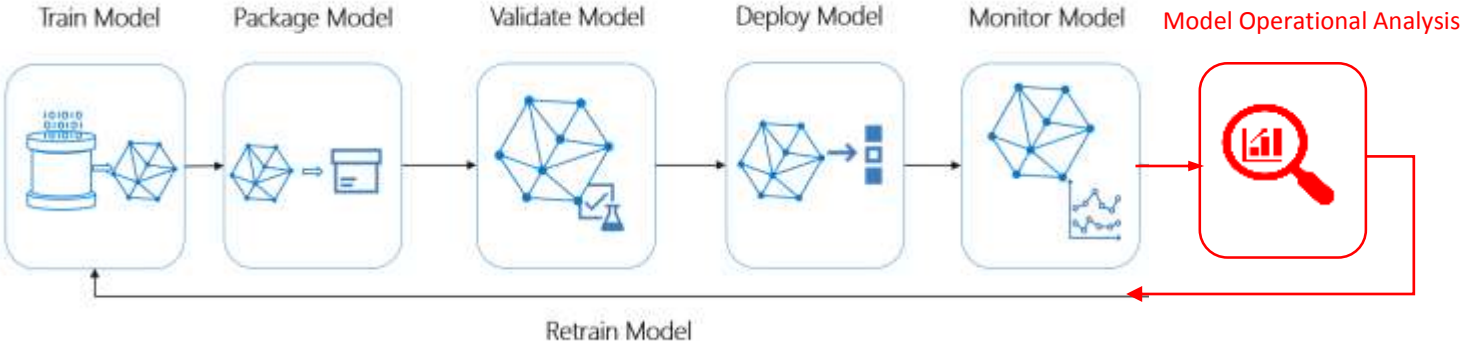
Goal – A Resilient DevSecOps System



SEI Modeling Collaboration



SEI Collaboration: Integrate the Analyses performed by the Data Scientist into the MLOps pipeline

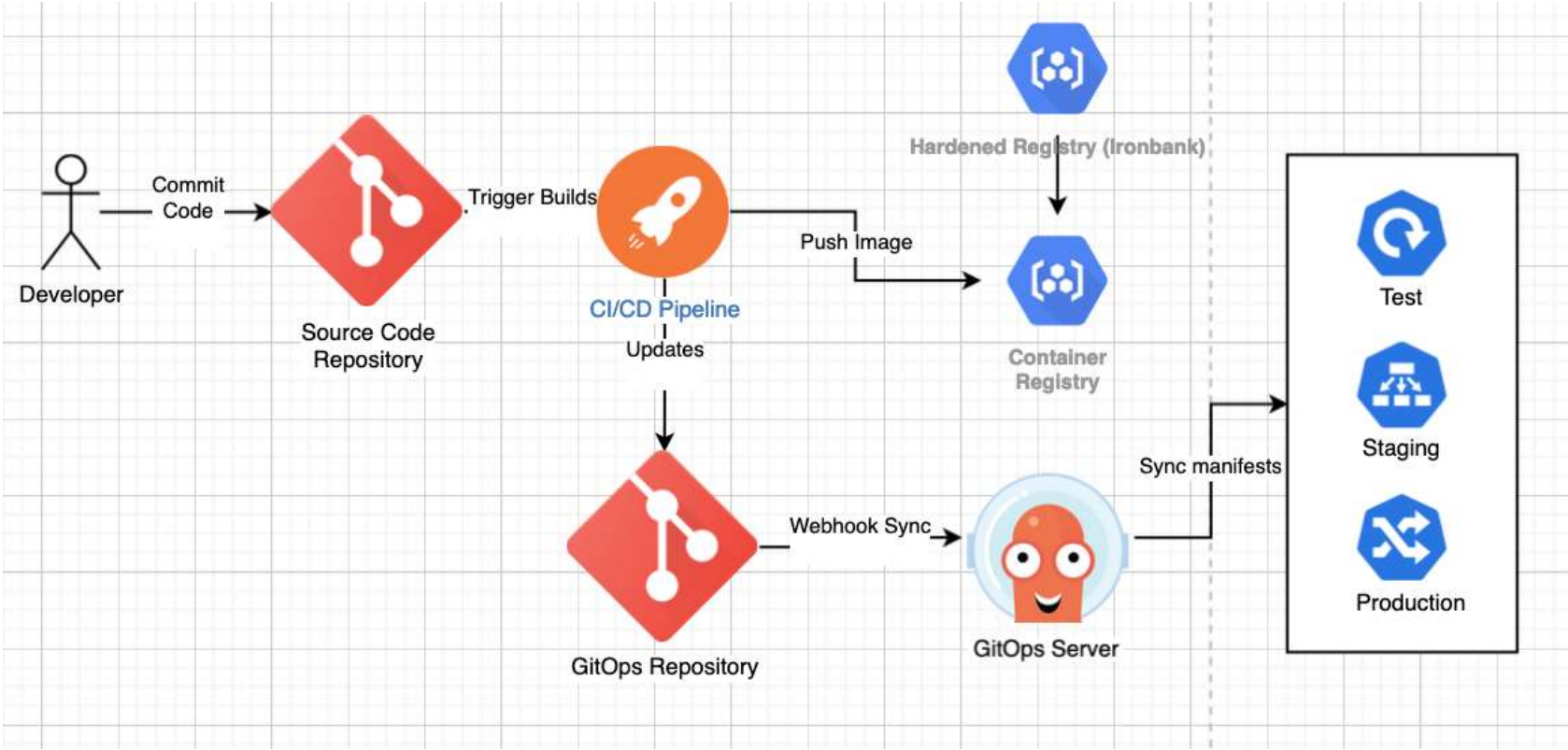


Model Operational Analysis should perform the first three steps of the model retraining process

1. [Analyze] Statistical analysis between the production data and development data
2. [Audit] Audit model performance
3. [Select] Integration of development and production data into a new development data set, with weights

Diagram Adapted from MS Azure MLOps Pipeline

Design approaches: Accelerated deployment



GitOps

- Configuration of applications and deployment environment is declarative and version controlled.
- Application deployment and lifecycle management is simple, automated, and auditable.
- Application deployments are fast, reliable, and idempotent.
- Deviations from version controlled configuration are detected and immediately remediated.
- Rollbacks are just deploying a different configuration

GitOps provides a mechanism to quickly change …. but

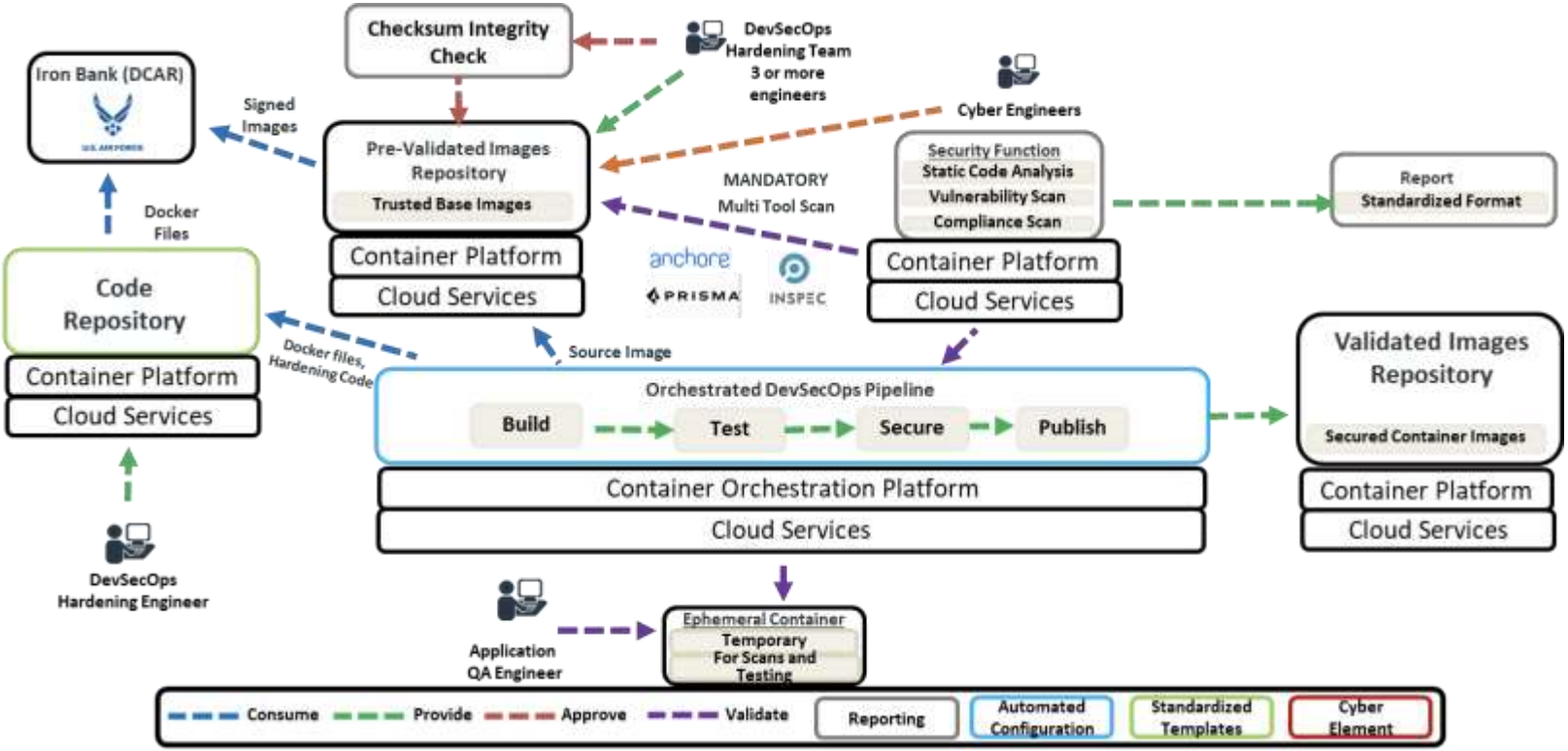
- Only when a container is built, it is deployed
- Need to have an event that forces a new container to be built based off of metrics you've identified and are monitoring

Developing use cases for metrics

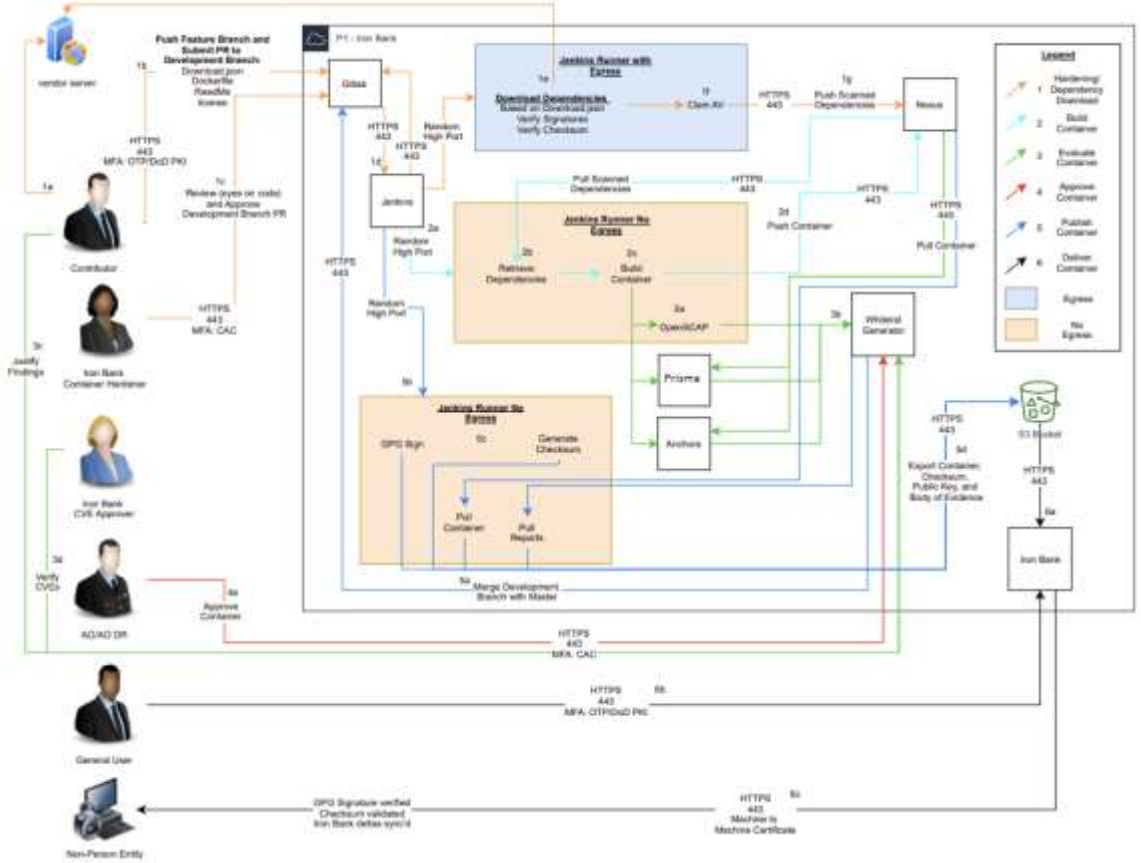
Consider a particular cyber resiliency metric

- Known versions of software libraries.
- How is this data collected? **Currently generated through container scanning.**
- Where is this data stored? **It is an artifact from the Iron Bank hardening process.**
- What can we do with this metric?
  - When a vulnerability is discovered in a software, a resilient system will identify:
    - If it exists in the system
    - Where it exists in the system
    - Take an action to evaluate, and mitigate a threat
- How is the ResEng project going to address this resiliency metric? **Monitoring is key.**
  - Requires queryable service that stores versions and locations of software libraries throughout the system and the deployed applications.
  - Requires a regular update mechanism to feed new vulnerability information

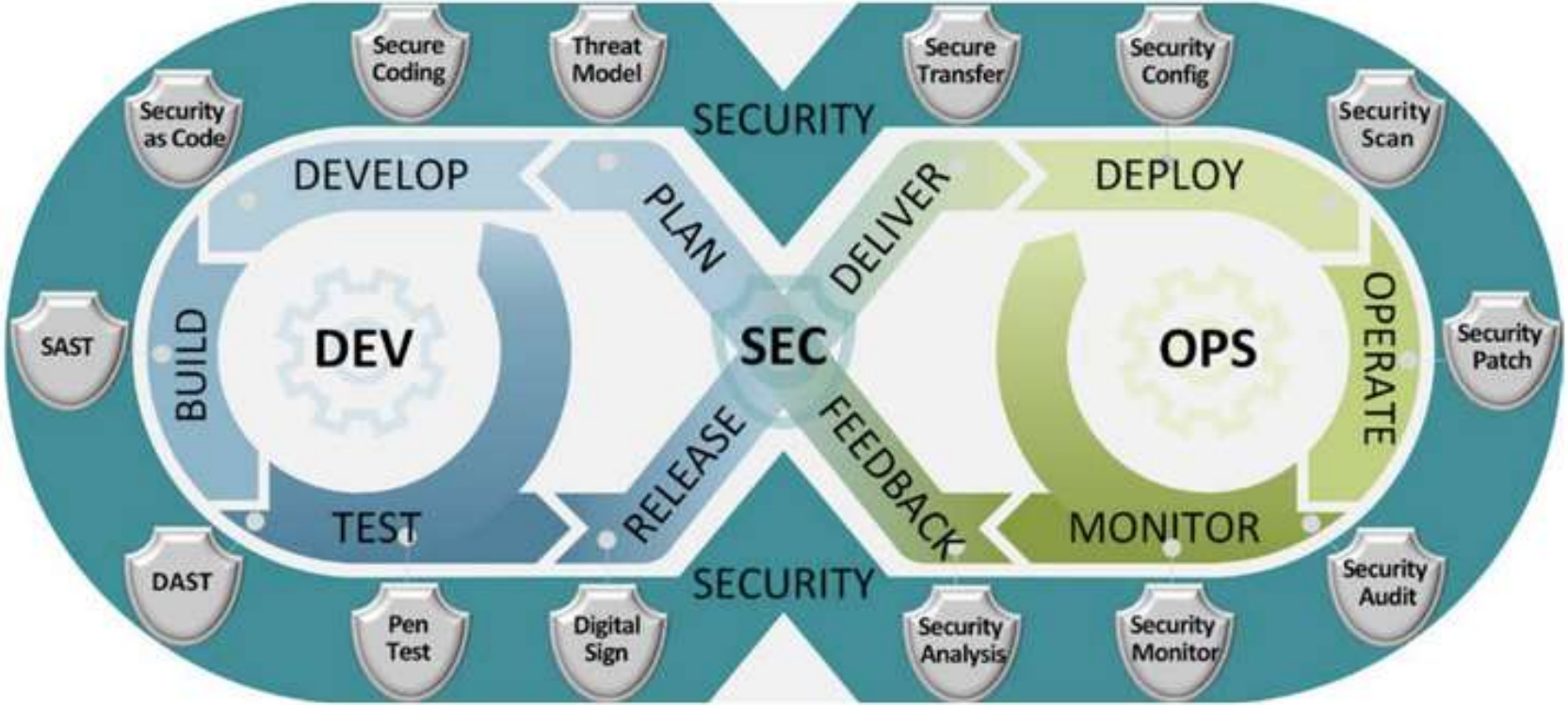
Iron Bank integration



Iron Bank Flow Diagram



Each step produces metrics, many must be continuously monitored for resilience.



Vulnerability scan

- Just 1 step in a DevSecOps pipeline
- At a point in time awareness of existing vulnerabilities present in your system:
  - How they are mitigated
  - Severity, and ultimately a ‘score’ to feed into overall system risk
- This can be a very complicated step with many artifacts
- This can integrated as an external service
- **This absolutely needs monitoring for new vulnerability information**

Resilience Metrics

Example of Infrastructure metrics (k8s)

Metric:	Service integrity checks performed on operational systems
Risk Category	Infrastructure – Continuity
Associated Risk Statement	If service integrity checks are not performed on operational systems, then the System integrity may be compromised leading to failure.
System CM Data Sources	Number of containers running, Number of containers available, Cloud (AWS, Azure, etc.) metrics (e.g. CloudWatch), Container Security (Clair, Anchore, etc.) scan results, known vulnerabilities
Threshold for Threat Analysis	<100% “Yes” for all containers

Example of Data/Application metrics

Metric:	Data associated with ML models used in both DevSecOps pipeline and deployed application(s).
Risk Category	Data/Application (eg. AI/ML Models)
Associated Risk Statement	AI/ML models must be monitored to assess risks associated with performance, data integrity, trend changes, etc.
System CM Data Sources	**Adjudication** • % correct/incorrect, true/false positives and negatives • classifier performance with alternate classifiers, changes over time **Changes to baseline distribution of raw data** • Keywords, # incomplete or erroneous elements, data drift **Algorithm** • Throughput, delay, memory consumption **Training data sources** • Availability of data sources, Reliability, Versioning, Reproducibility

Example of Development / DevSecOps pipeline metrics

Metric:	Metrics associated the development process.
Risk Category	Infrastructure/Cyber, development/deployment
Associated Risk Statement	Many metrics specific to development and maintenance can be aggregated to provide associated risk of an application.

System CM Data Sources	**Development process**
	• Mean-Time-to-Detect
	• Mean-Time-to-Respond
	• Mean-Time-to-Deploy
	• Identify higher risk software components
	  • Issues/Bugs per component
	  • Component with most defects

Additional Example Resilience Metrics to analyze

Must consider Quality, Cost, Accessibility, Availability, and Risk

Continuity/Utilization

- Time between decision to recreate resources and completion of the process
- Degree of degradation of a specific mission-essential function
- Average frequency of switches to an alternative resource per unit time
- Percentage of resources for which configuration changes can be made dynamically
- Percentage of mission-critical cyber resources which are recovered from a backup

Additional Example Resilience Metrics

Cyber Understanding

- Percentage of unauthorized changes to row data in a database that are detected
- Number of attempted intrusions stopped at a network perimeter
- Percentage of managed systems checked for vulnerabilities in accordance with policy

Transformation

- Percentage of services which have been made non-persistent
- Percentage of data stores for which automated deletion has been implemented
- Percentage of system components which can be selectively isolated

Prevention

- Percentage of resources to which dynamic changes can be made
- Percentage of administrators who can administer both network and security component

Preparation

- Average time to back up
- Percentage of mission-critical data stores for which a gold copy is maintained

Visualizing Resiliency

- Use Cases
- Metrics
- Services
- Dashboard