Eight Components of Autonomous, Self-Healing System

Matt Kehret
Digital Technologies Chief Technology Officer
Director of Architecture and Emerging Technology

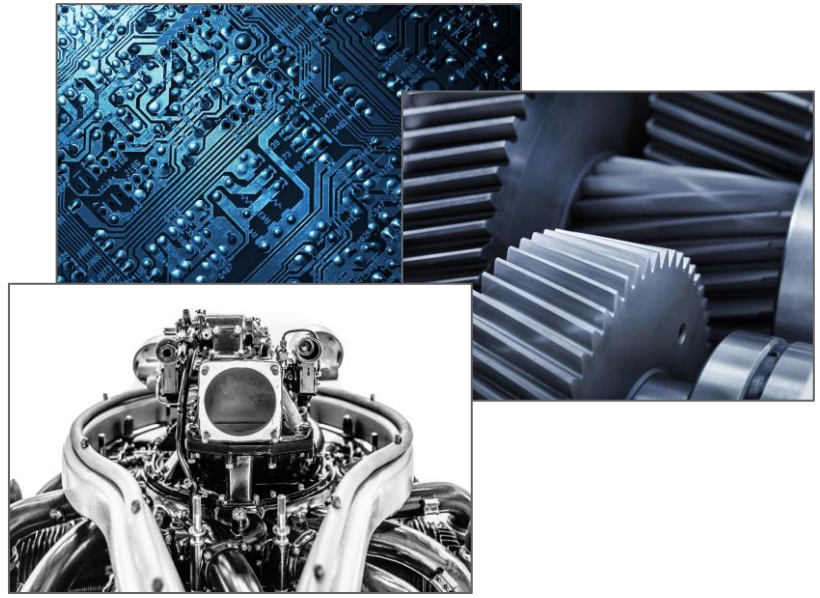
August 2022

Raytheon Missiles & Defense

Self-Healing Systems

- An autonomous, self-healing system can 1) detect when it is not performing as designed and 2) adjust its behavior to continue to deliver its intended value.

- Why create autonomous, self-healing systems?
  - Increase reliability and survivability; Proactive
  - Systems growing more complex
  - Failures happen
  - Required technology exists today

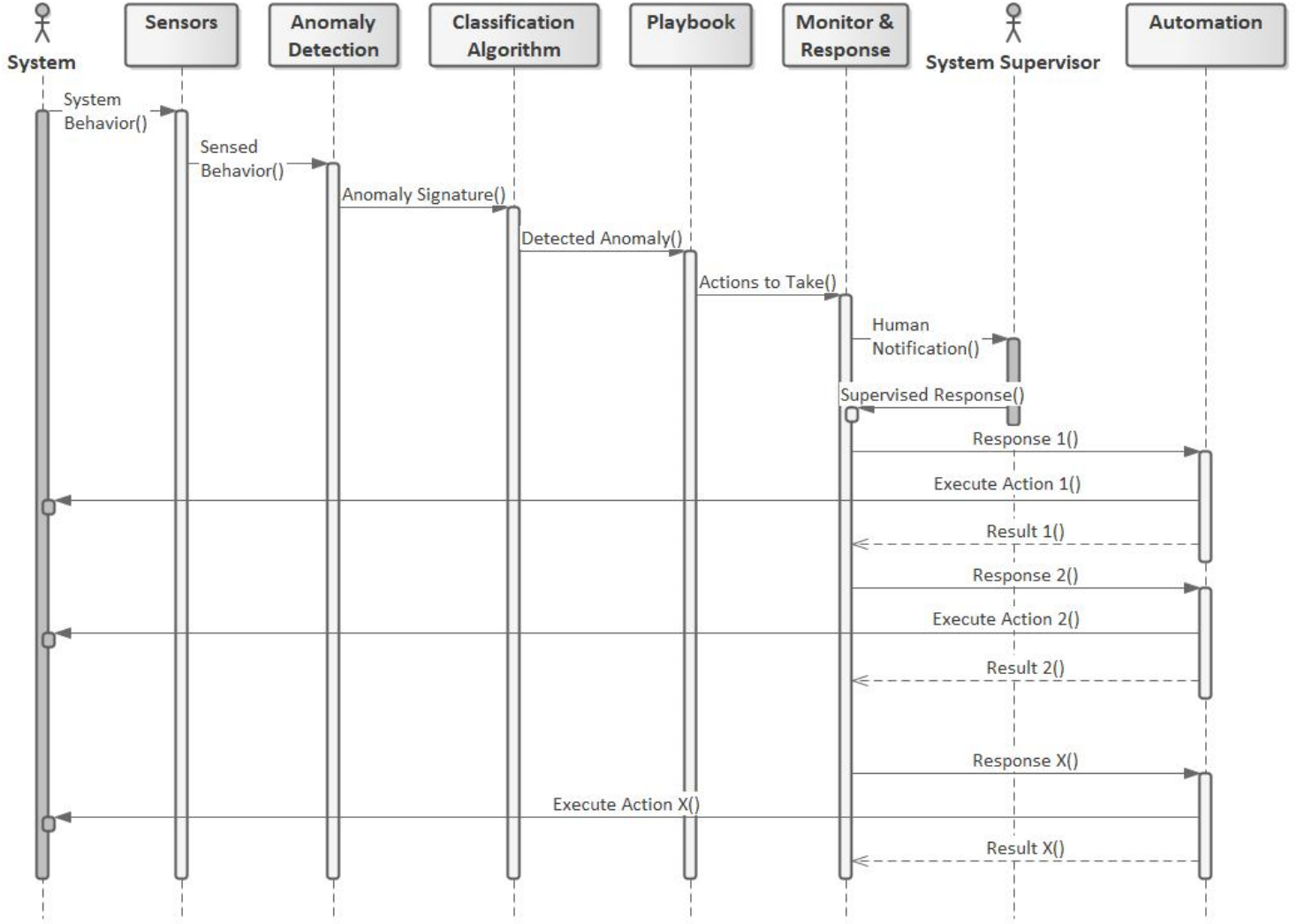


The pump is primed to deliver self-healing systems!

Companies with Self-Healing Systems

Components of an Autonomous, Self-Healing System

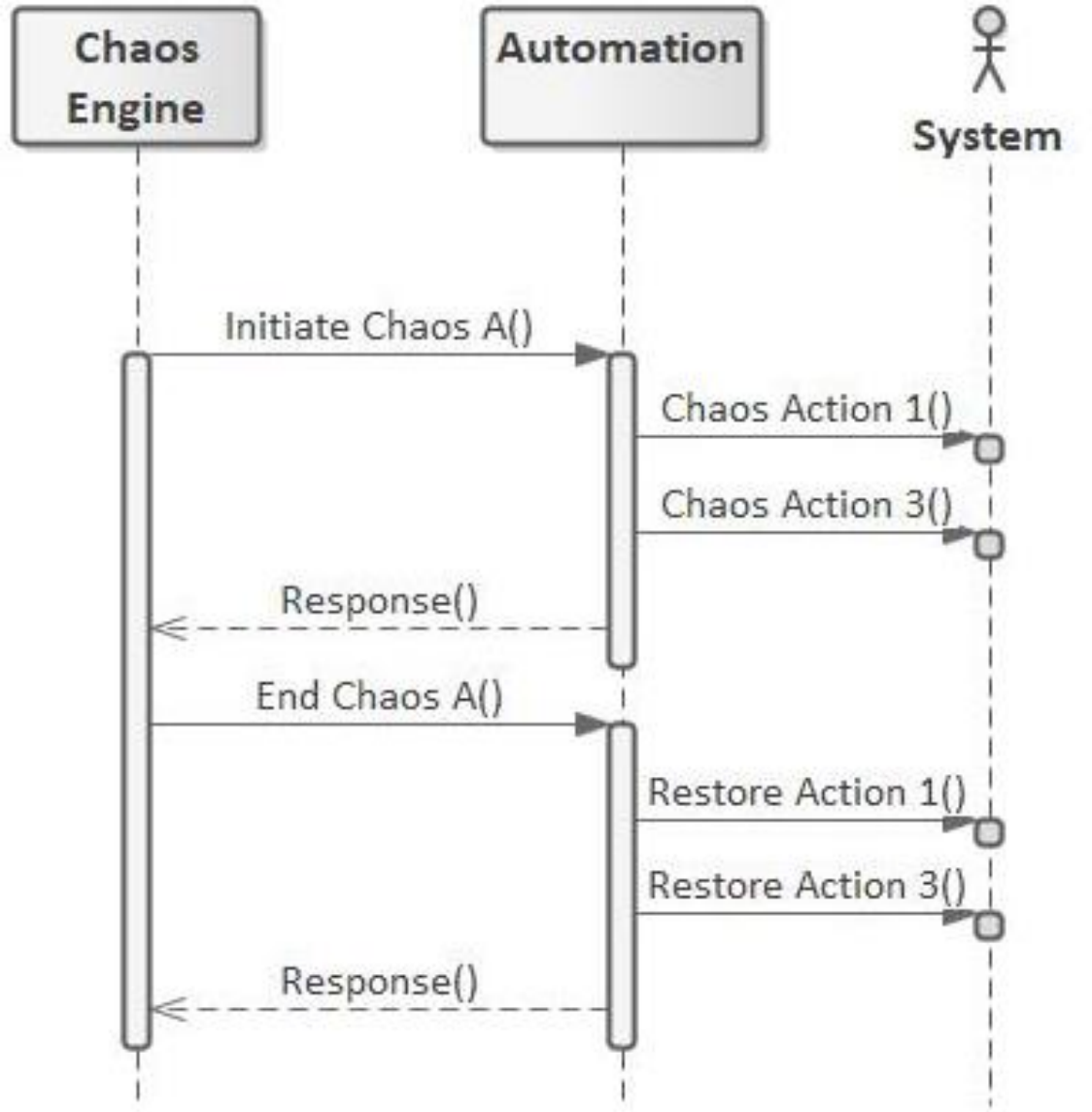
1. Sensors
2. Anomaly Detection
3. Classification Algorithms
4. Playbooks
5. Monitoring & Response
6. Automation
7. Engineered Chaos
8. Logging & Reporting



Known Technologies and Processes Exist to Create Each Component

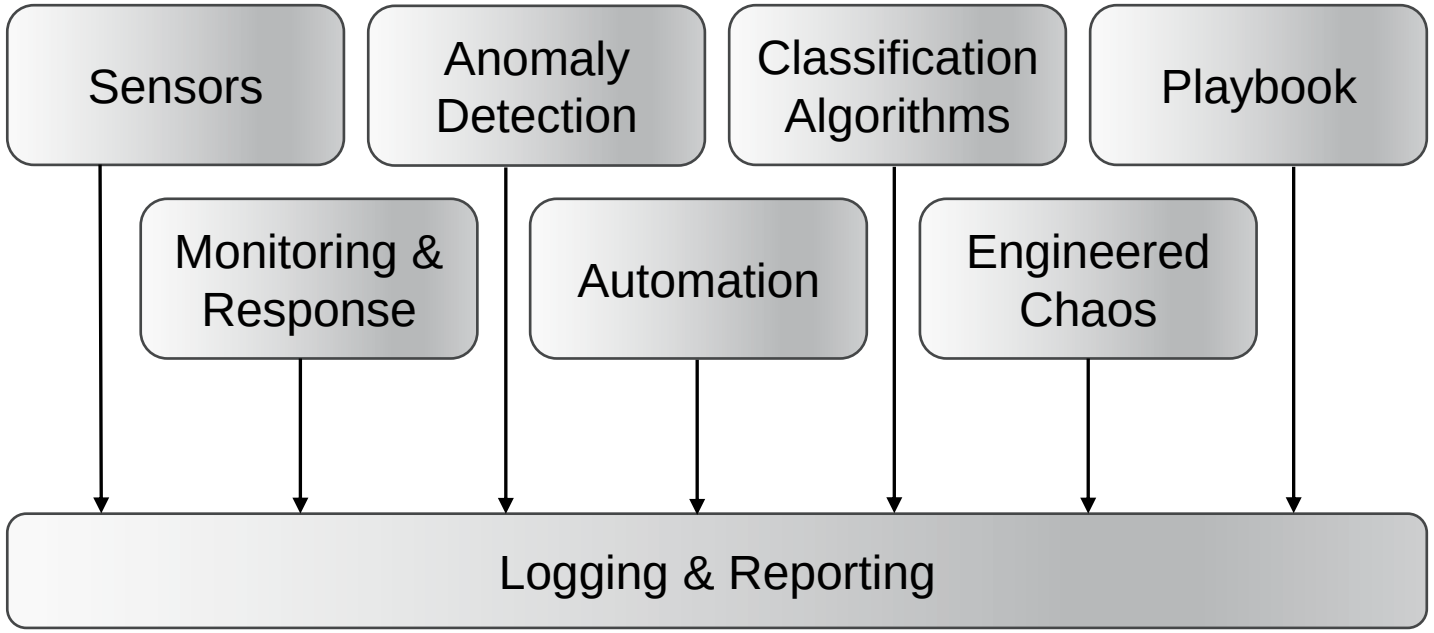
Engineered Chaos

1. Sensors
2. Anomaly Detection
3. Classification Algorithms
4. Playbooks
5. Monitoring & Response
6. Automation
7. **Engineered Chaos**
8. Logging & Reporting



Transparent Decisions and Actions

1. Sensors
2. Anomaly Detection
3. Classification Algorithms
4. Playbooks
5. Monitoring & Response
6. Automation
7. Engineered Chaos
8. Logging and Reporting



BONUS: Collaborative Communication

1. Sensors
2. Anomaly Detection
3. Classification Algorithms
4. Playbooks
5. Monitoring & Response
6. Automation
7. Engineered Chaos
8. Logging and Reporting

- Share environment and failures across system of systems
  - Swarm peers may be about to experience similar conditions
  - System of system peers may experience environment in different ways
  - Collaboration will help system to adapt and respond

Conclusion & Questions

- Logical Architectural Framework for building autonomous systems
- 8 key elements

1. Sensors
2. Anomaly Detection
3. Classification Algorithms
4. Playbooks
5. Monitoring & Response
6. Automation
7. Engineered Chaos
8. Logging & Reporting

- Questions?

We can make our products and systems more intelligent and autonomous today!

Thank you.

Component 1: Sensors

- Purpose
  - Capture data regarding the current behavior of the system.
- Technology
  - Digital, mechanical, hydraulic, etc devices capture system data
- Challenges
  - Anticipating reasonable failure points to detect
  - Balancing costs, both financial and computational, to capture system data

Component 2: Anomaly Detection

- Purpose
  - Process data captured by the sensors to determine if the system is operating within design tolerances. When system is outside of expected tolerances, send pertinent data to classification algorithms.
- Technology
  - Machine Learning algorithms are a prime way to detect anomalous behavior
- Challenges
  - Developing/training effective algorithms
  - Allocating sufficient processor cycles to perform calculations
- Examples
  - Unsupervised Machine Learning Algorithms
  - Statistical Approach

Component 3: Classification Algorithms

- Purpose
  - Process anomalous data provided by the anomaly detection algorithm and determine or classify the environmental challenge and/or failure. Pass the classification and pertinent data to the Playbook.
- Technology
  - Logistic Regression, Naïve Bayes, Decision Tree, Stochastic Gradient Decent, etc
- Challenges
  - Developing/training effective algorithms with proper detail
  - Allocating sufficient processor cycles to perform calculations
- Example Implementation
  - Yelp uses classification algorithms to compile, categorize, and label images

Component 4: Playbook

- Purpose
  - Take the identified failure condition and associated data and use it to determine the best course of action to respond.
- Technology
  - Lookup table, decision tree, machine learning algorithm
- Challenges
  - Anticipating failure states and determining best action for response
  - Developing sufficient data to train algorithms
  - Selecting appropriate responses to take given failure condition

Component 5: Monitoring and Response

- Purpose
  - When applicable, communicates with human in the loop to show diagnosed state and proposed response
  - Launches series of automated activities to carry out the desired actions to improve system performance in current environment. Monitors automation.
  - Tracks current health of system, performs system health assessments
  - Provides failure notification to system owners
- Technology
  - Software system and database
- Challenges
  - Effective interaction with human partner
  - Handling automation failures

Component 6: Automation

- Purpose
  - Enable the change of system behavior in response to anomalous behavior
- Technology
  - Alternative software, hardware, switches, etc
- Challenges
  - Anticipating useful automation
  - Avoiding over automation
  - Designing robust, redundant automation that works while parts of the system are failing
- Example Vendors
  - Chef, Puppet, Ansible

Component 7: Engineered Chaos

- Purpose
  - Intentionally introduces failures into production and development system. Encourages architects, designers, and builders to create autonomous systems. Enables the generation of data that represents failures.
- Technology
  - Software and hardware
- Challenges
  - Scoping the effort to include reasonable system failures
- Example Implementation
  - Netflix Chaos Monkey

Component 8: Logging & Reporting

- Purpose
  - Record all events taking place in the system to enable future analysis. Enables transparent system behavior and increases the understanding of the decisions made by the autonomous system. Provide the ability to report on the logged data in a way that users to connect events from the system components.
- Technology
  - Files, databases, software
- Challenges
  - Minimizing storage and processing requirements
  - Anticipating value added reporting needs
  - Capturing data from fielded systems
- Example Vendor
  - Splunk