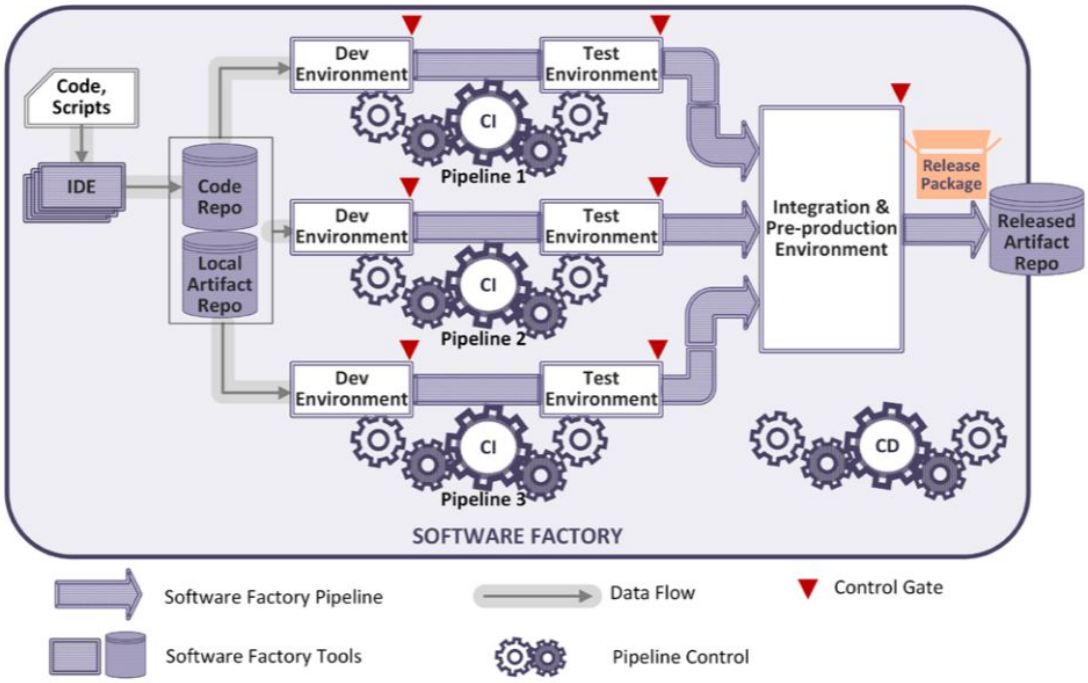


Provably Secure SDLC Processes Enable Frictionless Deployment of Smart Grids

How to Achieve Continuous Authority to Operate (ATO) Using a Disciplined Approach to Agile DevSecOps

By Saro Saravanan
Chief Strategy & Development Officer, Intuitus Corp. - https://intuituscyber.com
Strategy Advisor, The Agile Fractal Grid, Inc. - https://theagilefractalgrid.com

Power grids are considered to be part of a nation's critical infrastructure. The Agile Fractal Grid (AFG) has built a smart grid platform for modernization of aging electrical grids worldwide. Given the vulnerabilities of existing power infrastructure, AFG must ensure the highest levels of security baked into our smart grid platform, including the ability to detect and mitigate threats quickly. Additionally, to enable solution delivery at the speed of business, AFG is committed to meet the goals of the Digital Business Marketplace (DBM) for frictionless delivery of a basket of solutions from an ecosystem of partners. This paper provides an overview of the provable security process AFG uses to enable frictionless deployment of our smart grids.

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

Background/Acknowledgement

For electrical grids that could potentially be used by the U.S. military, the Department of Defense (DoD) provides the ATO. The traditional DoD ATO manual process is documentation-intensive, time-consuming, and expensive. The smallest code or configuration change, if its security impact is high, can result in several man-months of effort and several calendar months of delays to get into production. Recognizing its impact on agility, the DoD published methodology and processes in 2019 for a new continuous ATO (cATO) process that incorporates a DevSecOps reference design. DevSecOps is an organizational software engineering culture and practice that aims to unify software development (Dev), security (Sec) and operations (Ops).

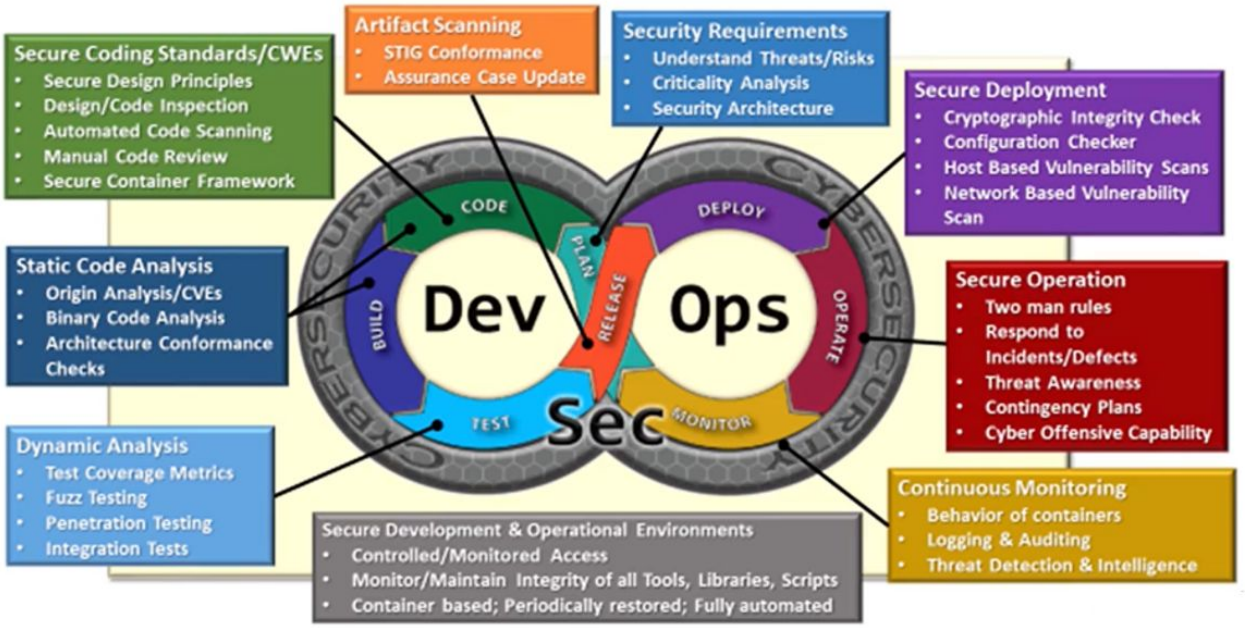
Mr. Nicolas M. Chaillan, Chief Software Officer for the U.S. Air Force, has been a champion of this process and is working relentlessly to enable wide adoption within the DoD and the Defense Industrial Base. Mr. Chaillan, previously an Internet entrepreneur who founded and led several successful companies, has been inviting industry and commercial participation in contributing to, learning, and applying the tools and methodologies to enable frictionless deployment.

A majority of global Internet companies have already adopted these DevSecOps methodologies, about a quarter of them with a high degree of maturity. AFG has embraced DevSecOps in all our endeavors. Given that the DoD is our ATO authority, AFG is fully invested in following the DoD's approach and framework. This paper draws *generously - at times verbatim* - from Mr. Chaillan's and the DoD's published unclassified and non-copyrighted materials on DevSecOps (see References).

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

Introduction

DevSecOps is an organizational software engineering culture and practice that aims to unify software development (Dev), security (Sec) and operations (Ops). The main characteristic of DevSecOps is to automate, monitor, and apply security at all phases of the software lifecycle: plan, develop, build, test, release, deliver, deploy, operate, and monitor. In DevSecOps, testing and security are “shifted to the left” through automated unit, functional, integration, and security testing - this is a key DevSecOps differentiator since operational, security, and functional capabilities are tested and built simultaneously.

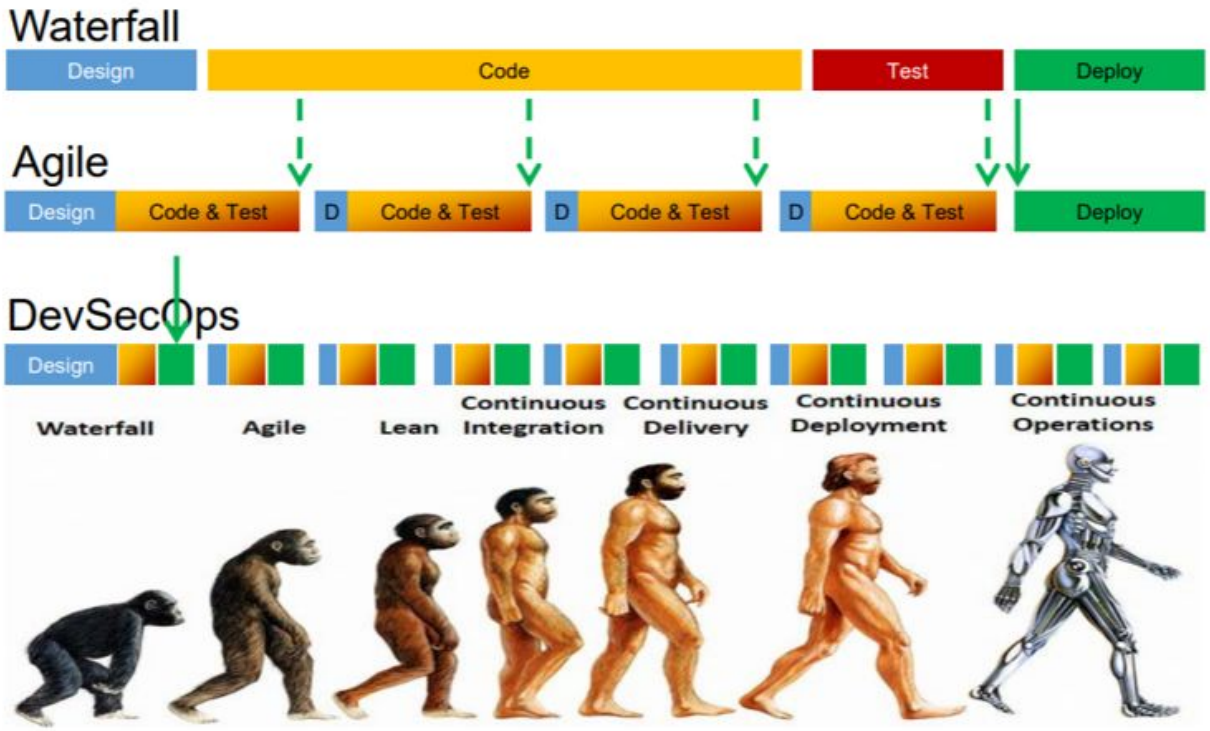


As opposed to the traditional model of agile development, where there is a separation of product development/engineering, operations, and security teams, the new model of DevSecOps demands a cultural shift where all teams have responsibilities in all aspects of the software development lifecycle. Security and operational considerations are built into every step of the development lifecycle by appropriate team members and highly automated.

The objective is to increase development velocity, operational agility, and security in equal measure to enable *continuous operations*, where features and bug fixes can be delivered into *production* in hours, not weeks or months. Manual configuration, settings, and deployment are completely eliminated. Infrastructure-as-code (IaC) and configuration-as-code (CaC) ensure provable running conditions. The source control and management system becomes a single source of truth for all aspects of software development, integration, testing, staging, and deployment. In effect, software is developed and delivered via a *software factory*, which provides a set of continuous delivery *pipelines* for teams to deliver software products and services.

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

This does not come easy. DevSecOps is not simply about deploying a set of automated tools, but requires rethinking and potential restructuring of the teams to ensure success. The DoD DevSecOps Reference Design document provides an excellent blueprint that bakes in best practices in a number of different disciplines within the software lifecycle. In this document, we will outline what development, security, and operational team considerations characterize DevSecOps.



Development Considerations

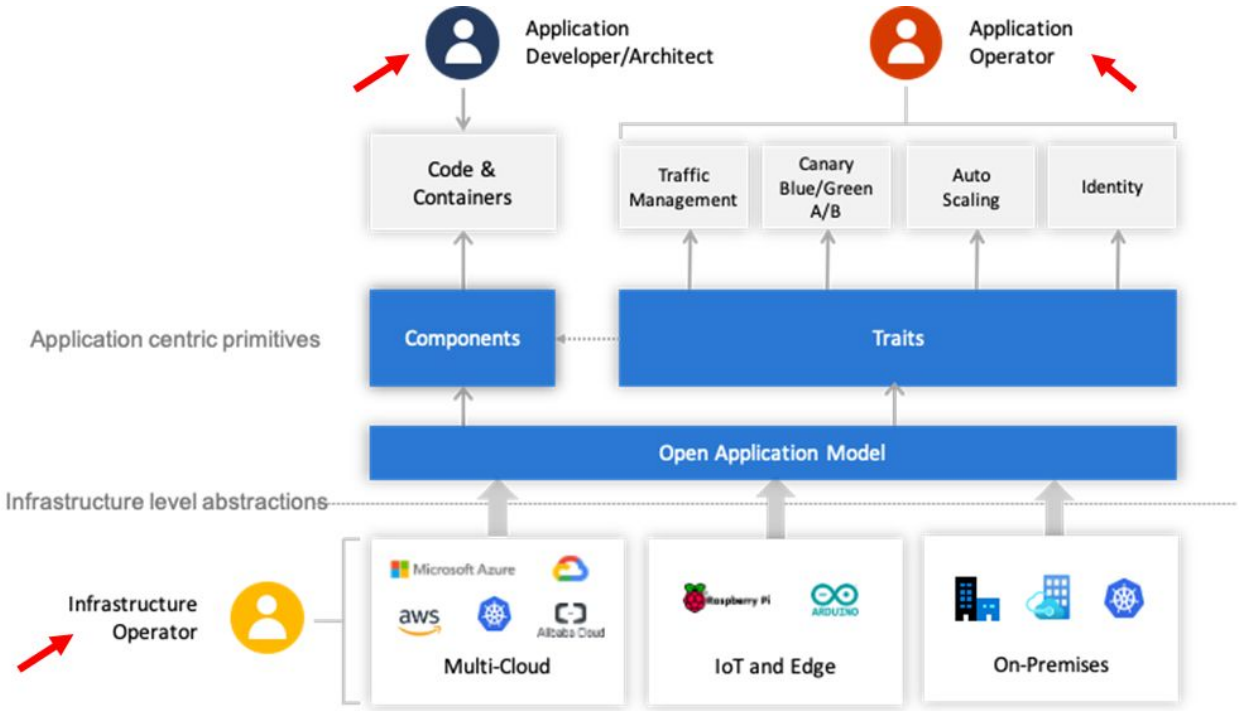
Containerization of applications and their deployment into Kubernetes clusters is now a widespread practice and provides the underpinnings of our model. The Cloud Native Computing Foundation (CNCF) fosters the growth and evolution of the ecosystem of open source tools that enable this model of computing. AFG will follow best practices that emerge from this ecosystem that are compatible with cATO guidelines.

Application developers in modern agile development teams are incentivized to learn the “full stack” and often find the need to worry about deployment and operational considerations. While this is an overall healthy process because we want all team members to have a good understanding of all concerns, there is an underlying lack of separation of concerns that, over time, has a detrimental effect on product quality and development velocity. See the mix of developer’s concerns and operational concerns in the following illustration of a Kubernetes YAML file, for example.

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

Source: https://www.cncf.io/blog/2020/07/24/open-application-model-carving-building-blocks-for-platforms/

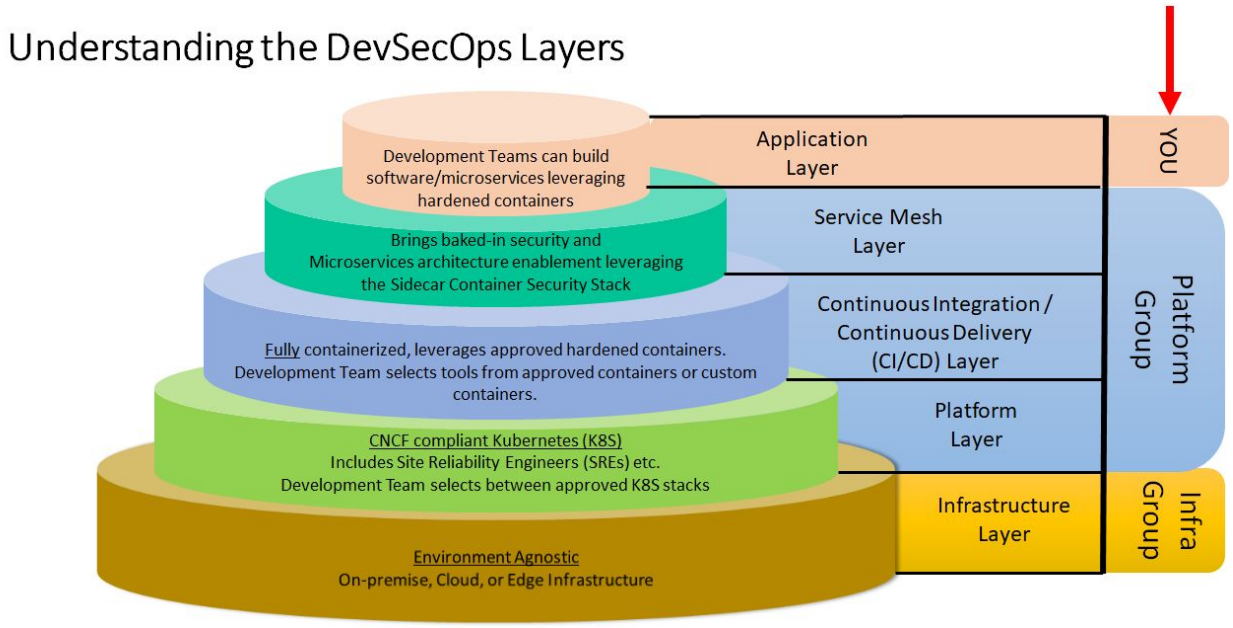
Ideally, a developer should only be concerned about implementing application features within the framework of the accepted architecture regardless of the environment in which the application would run. The Open Application Model (OAM) recognizes distinct roles for the Application Developer, Application Operator, and Infrastructure Operator, providing for a logical separation of concerns (see picture below).



This leads to a layered approach to DevSecOps adopted by AFG and DoD:

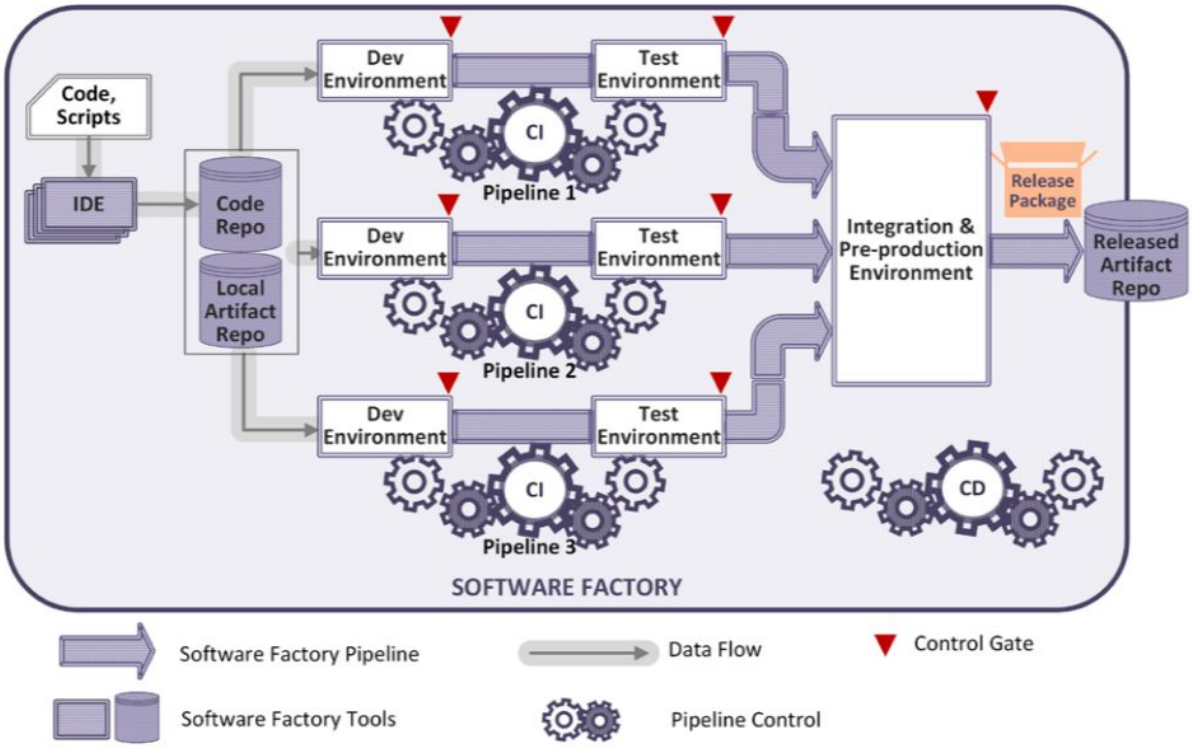
Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

- An infrastructure group can address the physical provisioning and security of on-premise, edge, or cloud systems;
- A platform group can provide an application platform-as-a-service that enables the establishment of one or more secure software factories as needed;
- Application developers can develop and deploy their applications in the factory.



A software factory contains multiple pipelines with a set of tools, process workflows, scripts, and environments, to produce a set of software deployable artifacts with minimal human intervention. It automates the activities in the develop, build, test, release, and deliver phases. The environments that are set up in the software factory should be orchestrated with scripts that include IaC and CaC and which run on various tools. A software factory must be designed for multi-tenancy and automate software production for multiple projects. An organization may need multiple pipelines for different types of software systems, such as web applications or embedded systems.

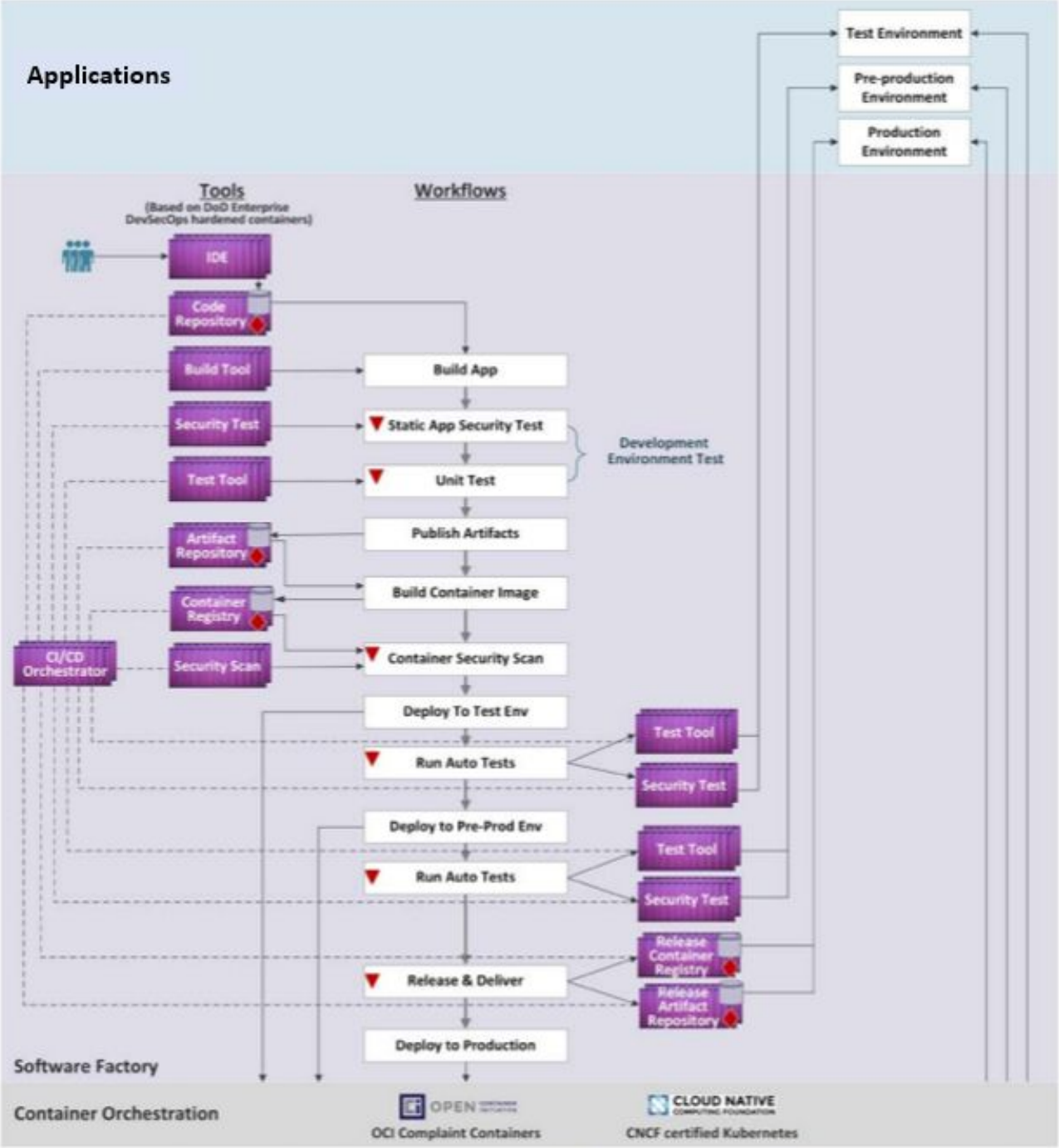
Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.



The AFG Compute team provides the infrastructure layer and platform layer to the application teams. The Compute team also helps the application teams set up software factories for their respective applications.

AFG's software factory reference design (shown in the picture below) includes the tools and process workflows to develop, build, test, secure, release, and deliver software applications for production deployment. All the tools are based on hardened containers. Committing code into the code repository kicks off the automated factory CI/CD pipeline workflow. The CI/CD orchestrator executes the workflow by coordinating different tools to perform various tasks. Some tasks are completed by a set of DevSecOps tools, such as build, static code analysis, unit test, publish artifacts, build container image, etc. Other tasks may need assistance from the underlying container orchestration layer, such as deploy applications to test, pre-production, and final production environments. Some test and security tasks may need human involvement.

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.



Put simply, quite a bit of the framework for DevSecOps comes from the adaptation of the DoD framework by the AFG Compute team for our partner application development teams. The AFG Compute team relies on the publicly available non-classified artifacts and infrastructure from the DoD DevSecOps program, which in turn relies heavily on CNCF-compliant open source and commercial software tools.

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

Security Considerations

The AFG Security team works closely with the Compute and Applications teams to help address end-to-end security considerations in every phase of the software development lifecycle, as outlined in the following table. Tool names are provided as examples only, no endorsement is implied; only tools from the DoD-approved list will be used by AFG to ensure cATO compliance. Note that the AFG Compute team has access to automated scripts to set up the full DevSecOps pipeline incorporating the chosen tools; they will work with the application development teams to set up each Software Factory.

Phase	Security Step	How Accomplished
Plan	Threat modeling to identify threats and vulnerabilities in the earliest stage of the development life cycle: Assets: What valuable data and equipment should be secured? Threats: What the attacker can do to the system? Vulnerabilities: What are the flaws in the system that can allow an attacker to realize a threat?	Tools such as OWASP Threat Dragon, Threat Spec, or Microsoft Threat Modeling used by the development team working with the security team.
Develop	Secure design principles and secure coding practices, e.g., as recommended by OWASP	IDE, Peer reviews, Pair programming. See references below.
Develop	Static code scan before commit	IDE security plugins such as Fortify, Veracode
Develop	Code commit scan	Source code repository security plugins such as SonarQube; secrets checking, e.g., git-secrets
Develop	Container or virtual machine hardening	Container security tool/Security compliance tool such as Twistlock

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

Develop	Software composition analysis scans for vulnerability checks on all the dependent libraries.	Tools such as OSS Review Toolkit or SonaType
Build	Static application security test and scan	SAST tool such as SonarQube or NodeJSScan
Build	Dependency vulnerability checking	Dependency checking / BOM checking tool such as OWASP Dependency Check
Test	Dynamic application security test and scan	DAST tool or IAST tool such as OWASP ZAP
Test	Manual security testing (such as penetration test)	Various tools and scripts (may include network security test tool) to be provided by Intuitus
Test	Container policy enforcement	Container policy enforcement tools such as Anchore
Deploy	Post-deployment security scan	Security compliance tools such as OpenSCAP
Monitor	System Security monitoring	Intuitus MDR+ CDAM

The following references can be used to learn more about secure design and secure coding principles:
https://wiki.sei.cmu.edu/confluence/display/seccode/Top+10+Secure+Coding+Practices
https://owasp.org/www-pdf-archive/OWASP_SCP_Quick_Reference_Guide_v2.pdf
https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04232020.pdf

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

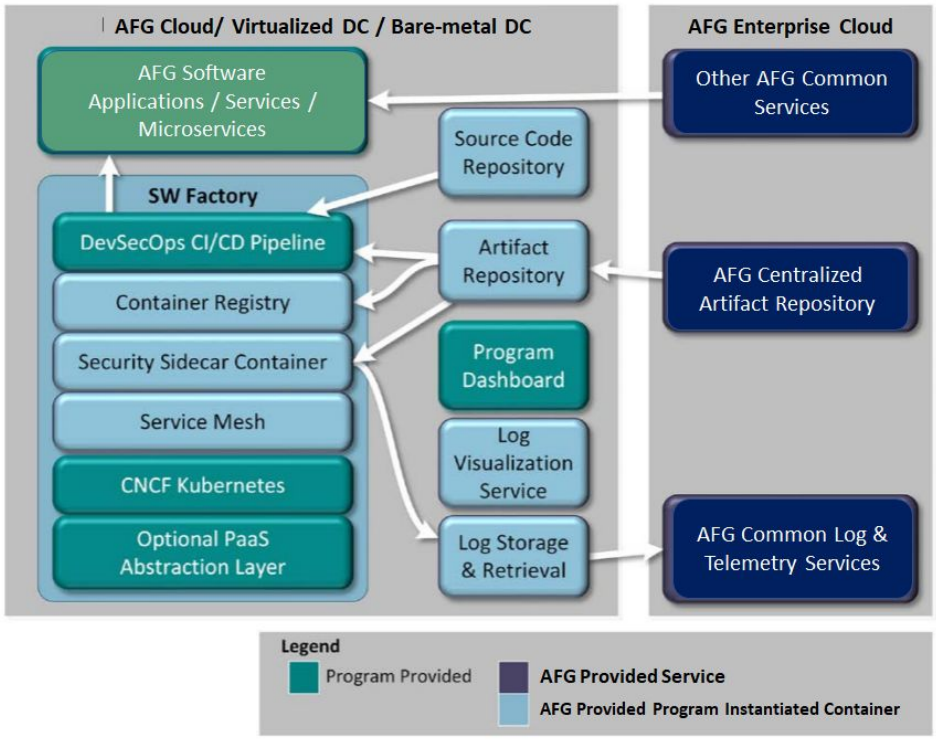
Baked-in Cybersecurity: DevSecOps Architecture & The Sidecar Container Security Stack

The picture below depicts the DevSecOps architecture:

1. Computing nodes of the deployment environment can be bare-metal servers or virtual servers in one of the approved cloud environments.
2. Applications are packaged into Kubernetes pods of Docker containers and deployed onto a CNCF-compliant Kubernetes system which allows the application to scale as prescribed within the set of nodes supported by the Kubernetes system.
3. On each Kubernetes node, elements of the Enterprise DevSecOps Platform, described later in this paper, implement the following:
    a. The Istio service mesh, which implements a zero-trust network, traffic management among microservices, and observability.
    b. Connection agent to the CI/CD pipeline, which performs updates to the running system based on signals from the source code management system, the single source of truth.
    c. A run-time defense artificial intelligence agent to perform behavior analysis on the application while it is running.
    d. A local container registry to keep the images pulled from central repositories.
    e. A logging storage and retrieval agent.
4. The connection agent pulls all required deployment artifacts from the centralized artifacts repository as needed.
5. Applications run as microservices within this architecture. Even legacy applications are packaged into microservices and replaced over time using the strangler pattern. (See Strangler Pattern.)
6. Within the same context as the Kubernetes pod in which the application runs, a Sidecar Container Security Stack (SCSS) runs side by side to proxy access to the DevSecOps platform.
7. Observability is built into the DevSecOps architecture because the logging agent uses Fluentd to push all pertinent logging information from the system and applications to a program-specific text and time-series database, from which centralized dashboards can observe and act.
8. Common Vulnerabilities and Exposures (CVE) Service / host-based security to provide CVEs for the security sidecar container.

The interaction of these services with the sidecar components is depicted in the figure below. The arrows show the direction of the data flow. The items in dark blue are services provided by the AFG, while light blue indicates components that are provided by the AFG but instantiated and operated by the program.

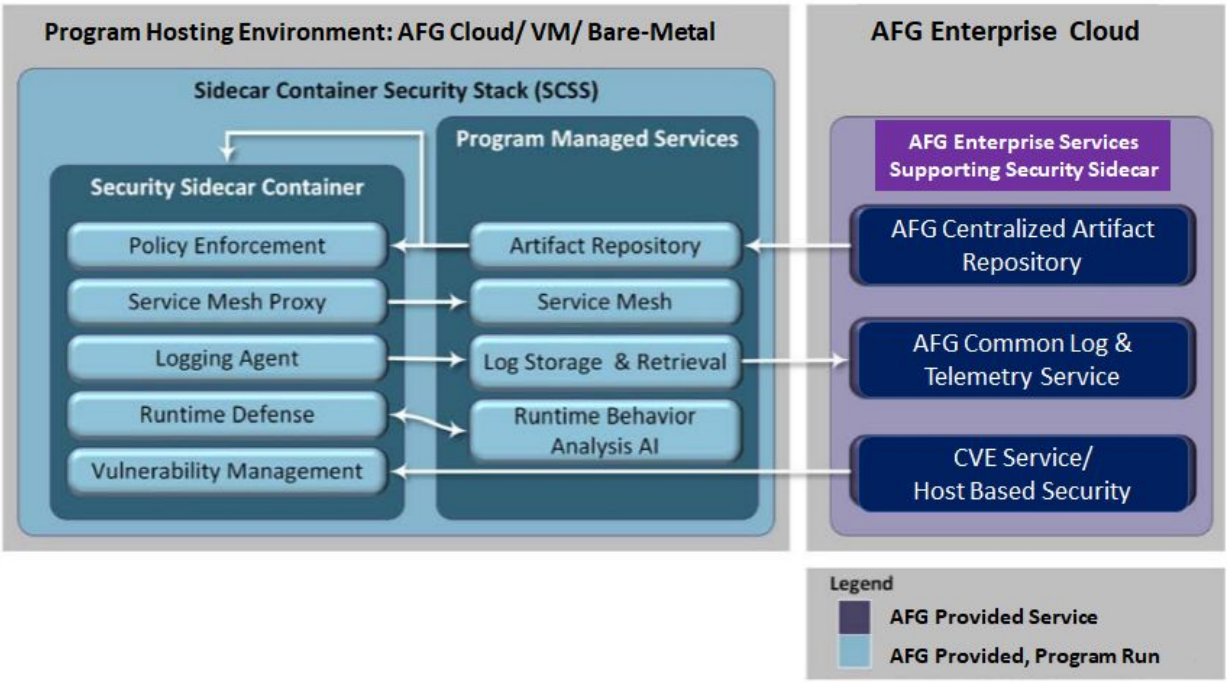
Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.



Sidecar Container Security Stack Interactions

A new service that is enabled by DevSecOps and the container-based Kubernetes runtime environment is the Sidecar Container Security Stack (SCSS). This security stack enables: correlated and centralized logs, container security, east/west traffic management, a zero-trust model, a whitelist, Role-Based Access Control (RBAC), continuous monitoring, signature-based continuous scanning using Common Vulnerabilities and Exposures (CVEs), runtime behavior analysis, and container policy enforcement. One advantage of using the SCSS is that Kubernetes can inject the sidecar automatically, without the team having to do anything, once it's configured.

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.



The Sidecar Container Security Stack is made of sidecar containers for a service mesh proxy and supporting services, including:

1. A logging agent to push logs to a platform centralized logging service.
2. Container policy enforcement. This includes ensuring container hardening from the hardened container registry is preserved and complies with the NIST 800-190 requirements.
3. Runtime Defense, this can perform both signature-based and behavior-based detection. This can also be used to send notifications when there is anomalous behavior.
4. Vulnerability Management
5. A service mesh proxy to connect to the service mesh
6. Zero Trust down to the container level. Zero trust requires strict controls, never trust anything by default and always verify. Key aspects of zero trust at the container level include mutual Transport Layer Security authentication (mTLS), an encrypted communication tunnel between containers, strong identities per Pod using certificates, and whitelisting rather than blacklisting.

Strangler Pattern

Martin Fowler, well-known software architect, describes the strangler pattern on his website. It is implemented as follows for a legacy application:

1. First, add a proxy, which sits between the legacy application and the user. Initially, this proxy doesn't do anything but pass all traffic, unmodified, to the application.

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

2. Then, add a new service (with its own database(s) and other supporting infrastructure) and link it to the proxy. Implement the first new page in this service. Then allow the proxy to serve traffic to that page (see below).
3. Add more pages, more functionality and potentially more services. Open up the proxy to the new pages and services. Repeat until all required functionality is handled by the new stack.
4. The monolith no longer serves traffic and can be switched off.

Secrets Management

Secrets such as credentials for various environments, private keys, TLS certificates, and so on must be held in vaults such as Hashicorp Vault or Ansible Vault. The Software Factory will enforce this.

Proactive Security Posture: Managed Detection & Response, Continuous Diagnostics & Automated Mitigation

In addition to employing best practices in DevSecOps to developing and deploying applications, AFG and partner companies need to have a provable proactive security posture as they also have operational responsibilities for most smart grids that are deployed by AFG. Intuitus Corp. is the cybersecurity and cyber physical systems security provider for AFG and the core part of the AFG Security team.

The AFG Security team provides enterprise-level common services that facilitate cybersecurity enforcement and IT management. One security service will perform traffic inspection and filtering to protect the mission enclave and mission applications. Some security service examples include firewalls; Intrusion Detection System (IDS)/Intrusion Prevention System (IPS); endpoint protection and malware detection; data loss prevention; host-based security; log/telemetry aggregation and analysis; and Identity, Credential, and Access Management (ICAM).

Intuitus Managed Detection and Response (MDR) is a fully self-contained, automated defense-grade cybersecurity solution monitored by human experts as an outsourced service at a very affordable price to small-to-medium sized businesses and is equally applicable to global enterprise entities.

There is a litany of tools and approaches for *action-oriented* approaches to securing your business, but the elements of an *outcome-oriented* approach are simple. To detect attacks early and thwart breaches before they happen, at a minimum, the following elements need to be in place:

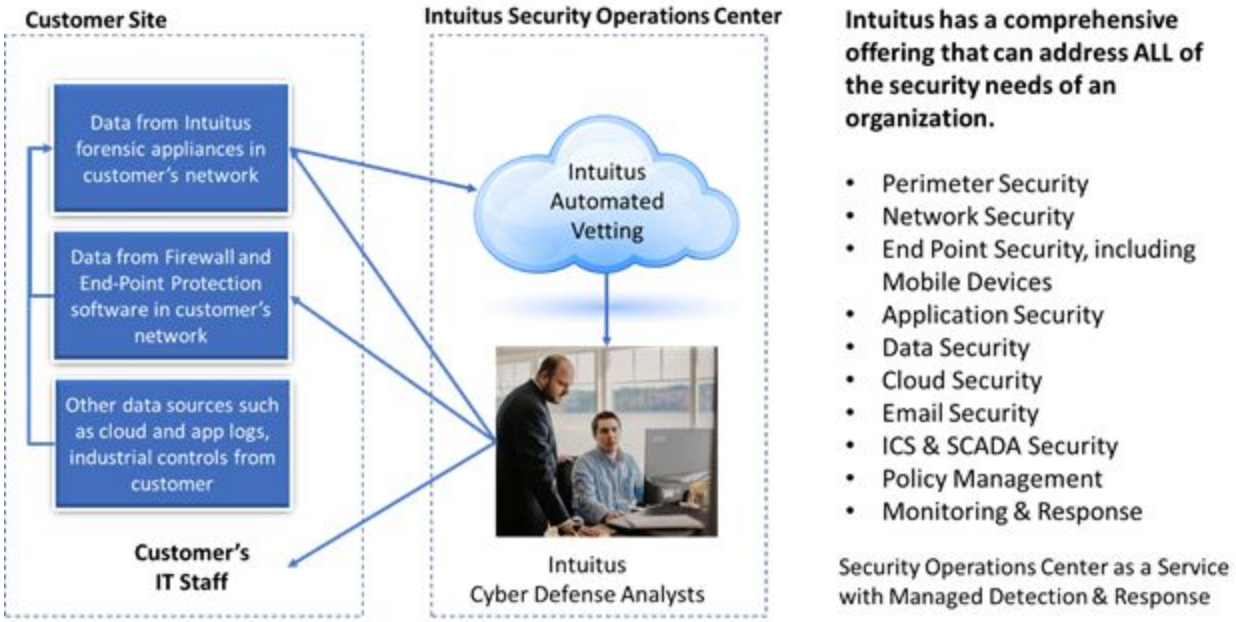
- Instrumentation of the organization's unique network segments and key host systems to capture all activity.

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

- Automated intelligent procedures to detect off-pattern and nefarious activity early.
- A security operations center (SOC) staffed 24/7 by trained cybersecurity analysts who can monitor unusual activity for possible breaches and employ other proactive measures such as threat-hunting.
- An incident response team and plan that can rapidly implement counter-measures for any detected nefarious activity.
- Continuous ongoing vigilance and training within the organization aimed at preventing infiltration.

Intuitus's solution can also be categorized as SECaaS (Security-as-a-Service) or SOCaaS (Security Operations Center-as-a-Service) depending up its deployment model. The core technology has been in continuous development since 2007 by The Boeing Company and Intuitus technologists and has been protecting some of our Nation's most vital assets for more than a decade. Intuitus has developed highly sophisticated security management appliances that work in tandem with Intuitus's global SOC based on this unified threat correlation technology. Intuitus's operations are conducted by a formidable team of cybercrime fighters, many of them veterans, dedicated to making the world safer from cyberattacks, 24/7/365.

Intuitus cybersecurity defense analysts and consultants can also provide comprehensive support to set up incident response plans and in all matters relating to cybersecurity governance, risk, and compliance.



Intuitus can eliminate a lot of costs and headaches while at the same time providing an outcome-based approach (as opposed to procedure-based) to cybersecurity and cyber-physical systems security. At the end of the day, it does not matter that your employees were trained in cybersecurity or that you followed procedures - all it takes is one weak link for hackers or insiders to exploit your network. With Intuitus, you will be alerted immediately and cyber defense

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

experts will help you respond effectively before any lasting damage can be done. The Intuitus methodology is so thorough that if you want to prosecute the perpetrators or provide information to Law Enforcement, all of the forensic evidence is preserved.

The latest innovation from Intuitus is an improvement on MDR, called Continuous Diagnostics and Automatic Mitigation (CDAM), which automatically mitigates threats on a near-real-time basis while coordinating with Intuitus's highly-trained cyber defense analysts. By implementing MDR with CDAM, an organization's security posture can be transformed to a proactive, active cyber defense stance very quickly, in a matter of days or weeks depending on the complexity of the networks.

Continuous Diagnostics & Automated Mitigation (CDAM) Example Scenario

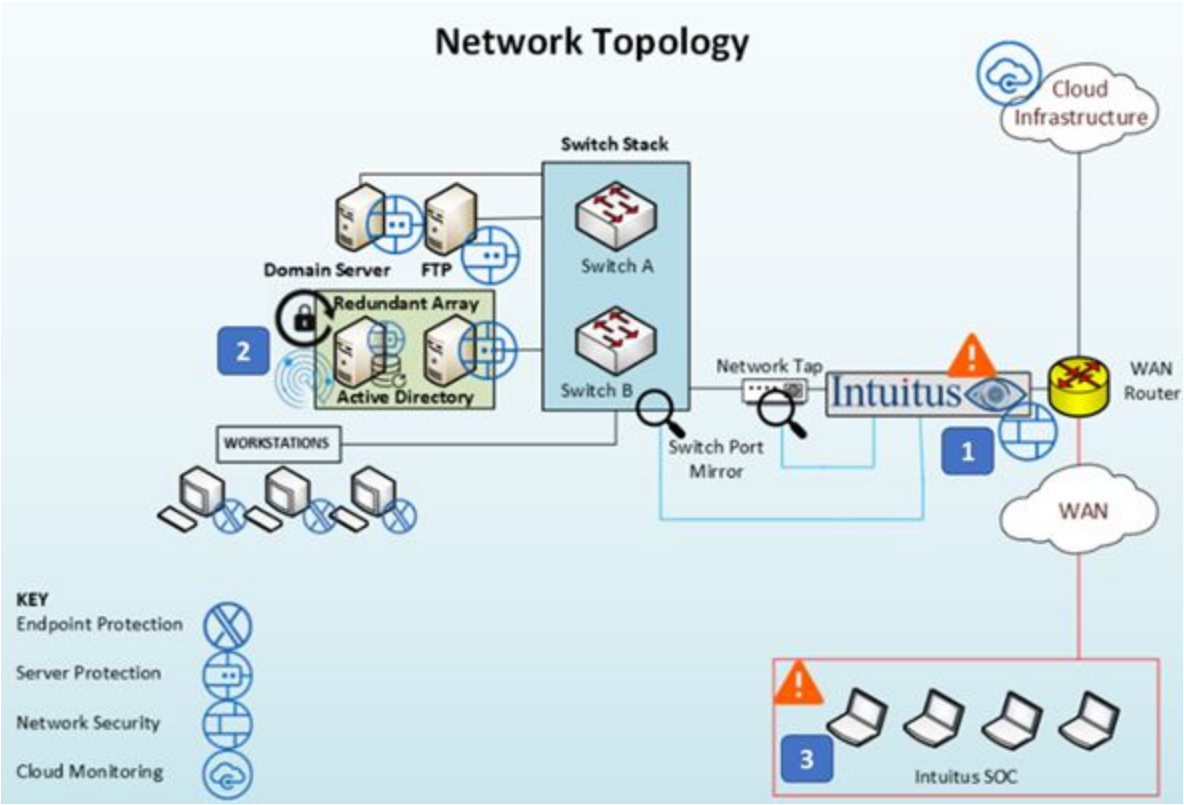
The picture below illustrates how Intuitus CDAM works. The following provides an explanation of the major players involved, labeled with numbers in the illustration:

1. Intuitus Security Management Appliance (SMA) which includes instances of forensic data controller, security utility server, and a firewall, along with necessary integrations.

2. A Microsoft Active Directory Controller node and its backup node in a redundant array configuration.

3. The Intuitus Security Operations Center.

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.



The example scenario unfolds as follows:

1. A client hires a third-party technical firm to perform routine maintenance on an on-prem server.
2. The third-party technical firm dispatches a technician to the client's location to perform the maintenance.
3. The technician arrives on site and connects removable media to the server to upload the maintenance software.
    a. Unknown to the technician, at a previous service call, her removable media was infected with a zero-day ransomware payload.
4. The technician performs the required maintenance and leaves the data center.
    a. The ransomware payload on the technician's removable media discretely uploaded itself to the server during the maintenance. It will begin encryption at midnight of the day following the maintenance.
5. At midnight of the day following the maintenance, the zero-day ransomware payload begins encryption on the server and simultaneously begins scanning the network for other targets.
6. The Intuitus server protection agent recognizes the encryption and begins caching each file and directory being encrypted by the ransomware payload. It simultaneously notifies the Intuitus SMA of the encryption.

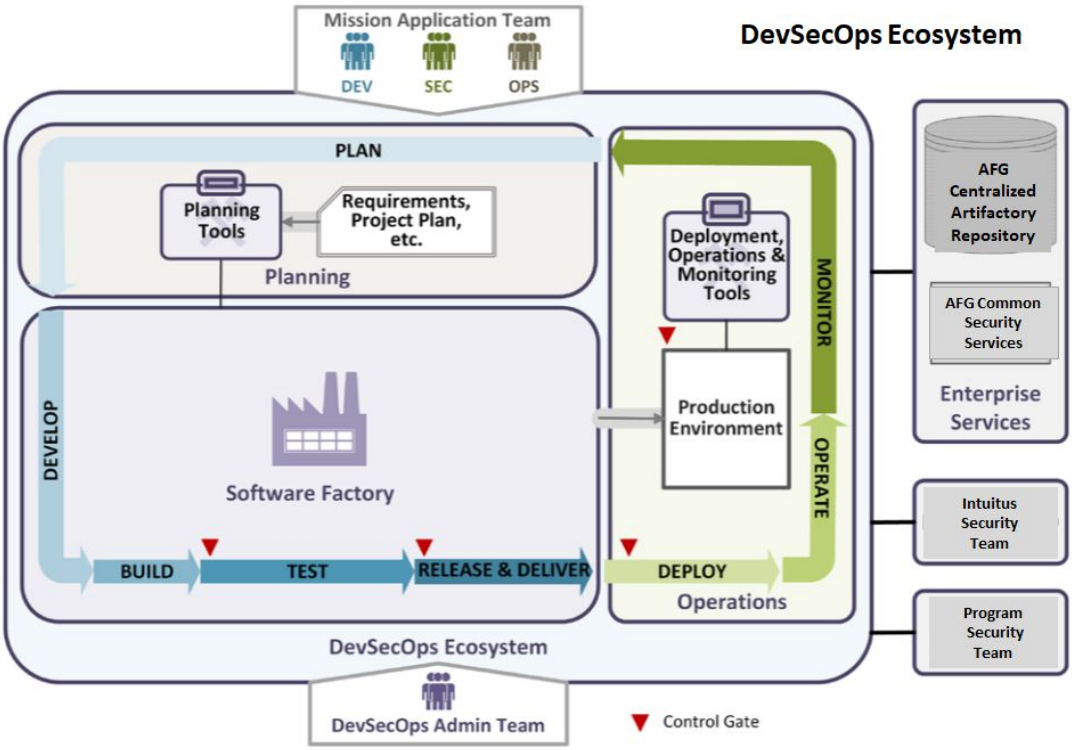
Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

7. The Intuitus SMA correlates the encryption activity with the network scanning activity it observed via network monitoring points and asses the server to have a high probability of compromise.
8. The Intuitus SMA instructs the server protection agent to quarantine the payload and isolate the server from the network using the server's routing configurations and host firewall settings.
9. The Intuitus SMA informs other endpoints on the network to refuse any traffic from the affected server until instructed otherwise.
10. The server protection agent isolates the affected server and only allows it to communicate with the SMA. It also quarantines the ransomware payload and begins recovering encrypted data from its cache.
11. Once recovered, the protection agent initiates a full system scan and notifies the SMA when the server is verifiably clean and the threat has been remediated.
12. The SMA initiates system scans on other network endpoints to verify that they have not been compromised and instructs the server protection agent to restore network access for the previously affected server.
13. The server comes back online and resumes its normal functions with no data loss and very minimal down time.
14. The SOC analysts have been monitoring the activity throughout the process and are analyzing the data to determine the root cause, notify the customer of the breach, activate the incident response plan, and report the sample for signature generation and future detection.

Operational Considerations

The following figure shows the DevSecOps Ecosystem. The Mission Application Team refers to team members of one major product or service function, e.g., Electrical Power. Teams comprise members with skills in each of the major areas of development, security, and operations pertaining to the domain in question.

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

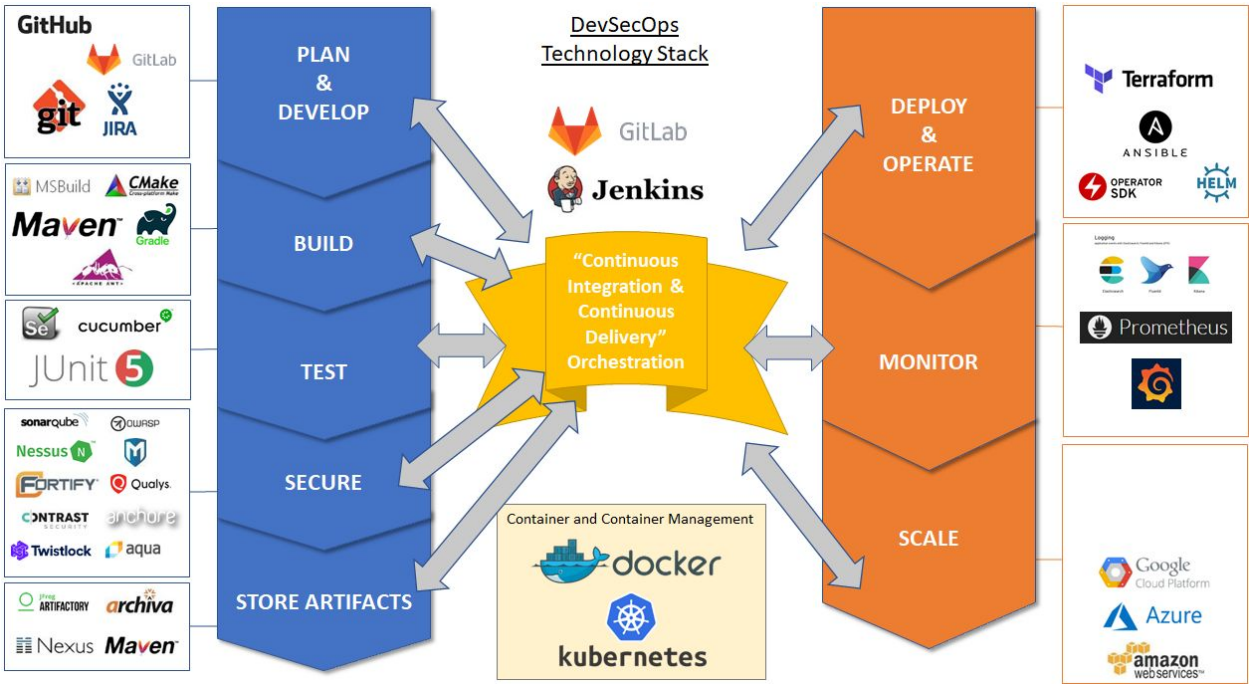


Standing Up the Infrastructure

Continuous Integration/Continuous Deployment Pipeline

DoD has a reference implementation of the CI/CD implementation on Amazon Web Services (AWS) GovCloud. DoD maintains source code for this as well as many of the ecosystem artifacts at a publicly available website. AFG is adapting the reference implementation for our CI/CD pipeline. The following figure provides an overview of the reference implementation and the choice of approved products for various aspects of the SDLC.

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

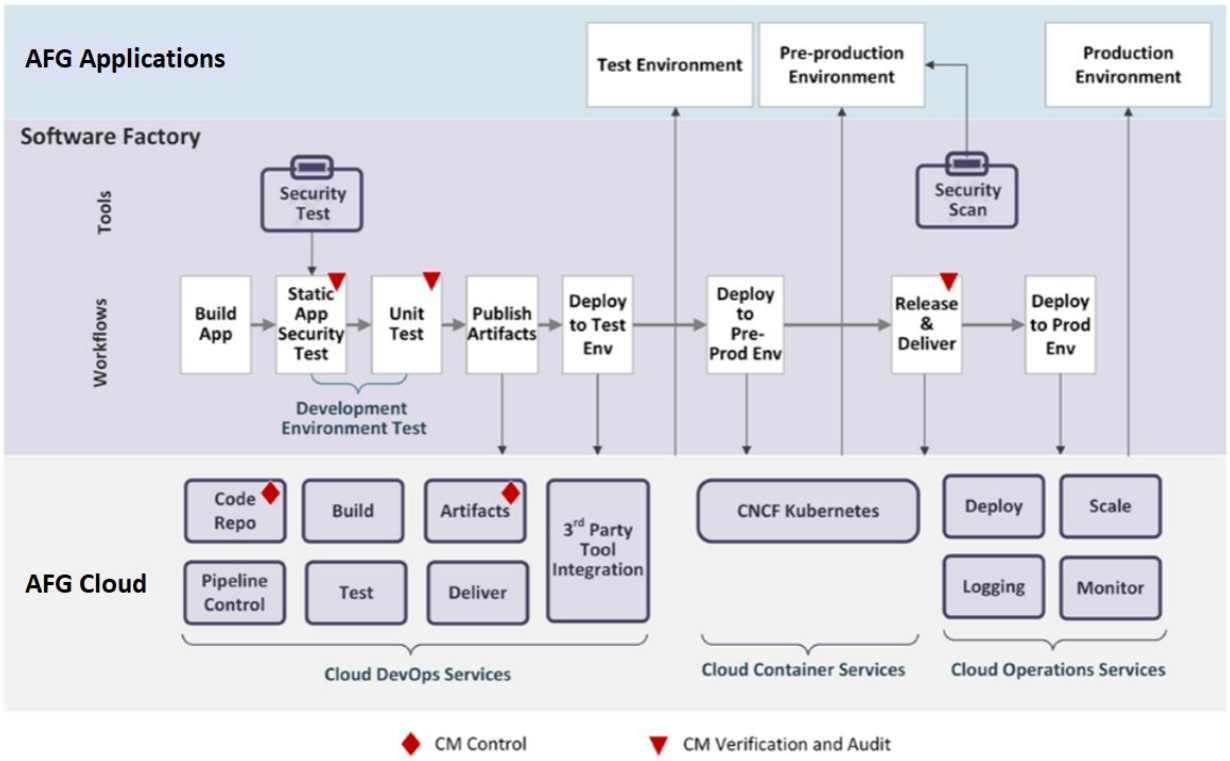


A full and growing list of approved products is maintained by the AFG Compute team as a subset of the DoD-approved products. The entire CI/CD environment can be stood up using IaC and CaC scripts by the AFG Compute team.

Deployment Target Environments

The following figure shows the setup to be run by the AFG Compute team. The AFG Cloud hosts all common artifacts pertaining to the DevSecOps services, the cloud container services, and the cloud operation services. The Software Factory uses these artifacts as part of its pipelines to produce the applications and services. Applications run on the target environments according to their desired characteristics. For example, the test and pre-production environments could run in a private data center and the production environment would be Tier-1 for edge applications, Tier-2 for district-level applications, Tier-3 for state-level, and Tier-4 at the national or international level.

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.



Argo CD, a declarative GitOps continuous delivery tool for Kubernetes, runs on the target environments to ensure that updates are promptly “pulled” into the targets consistent with the truth in the source control management system (Git).

Identity, Credentialing & Access Management (ICAM)

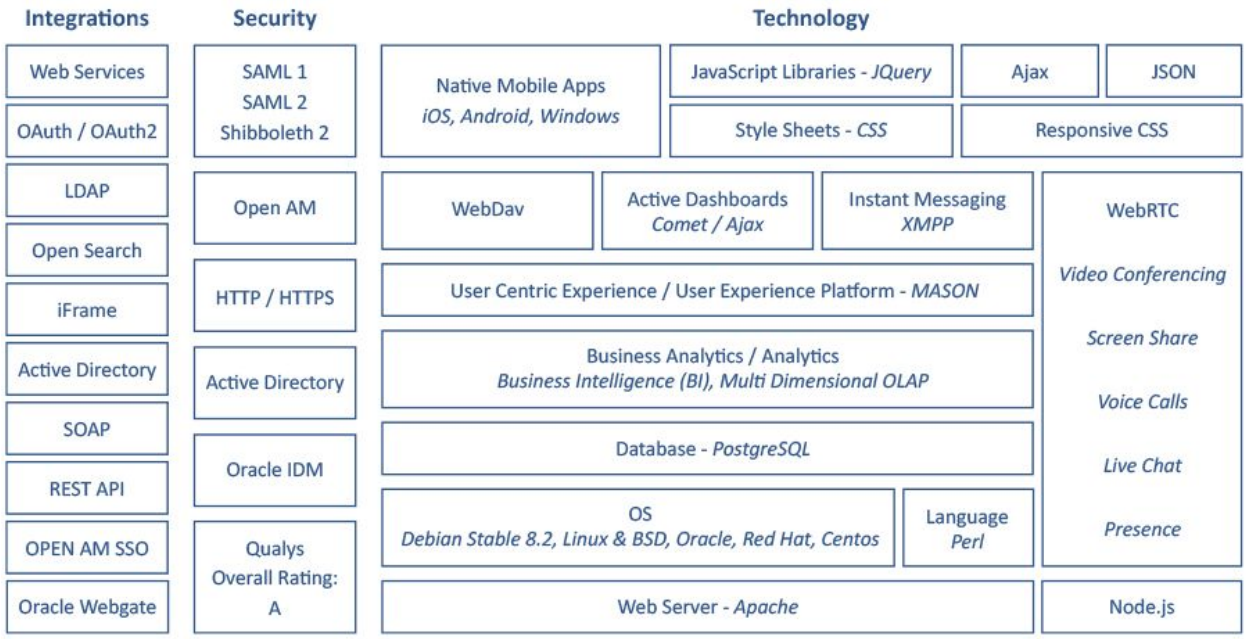
Any entities seeking access to any of the controlled environments must be appropriately credentialed and access strictly controlled according to authorization. The following elements together constitute the AFG ICAM framework:

- Comply-to-Connect (C2C) - an automated network access control solution that verifies an endpoint is authorized and meets security requirements *before* granting access to the network.
- Intuitus Continuous Diagnostics and Automated Mitigation (CDAM) with the Security Management Appliance (SMA), which can enforce C2C and other policies.
- Keycloak Provides Single-Sign On, SAML, OpenID, Oauth, LDAP / AD Integration with Internal and External Identity Providers, including multi-factor authentication (MFA). Keycloak is generally within the confines of AFG-maintained internal solutions (tiers 1, 2, and 3).
- MVine Identity Broker (see picture below) solves the issues faced with multiple heterogeneous user stores or cross domain user provisioning by connecting multiple Identity Providers (IdPs) and Service Providers (SPs) via a SAML2 IdP proxy, primarily for AFG-external solutions (tier 4). It is the perfect solution for businesses with complex

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

user identity and application management requirements that span multiple business units, departments, and partners.

- Multi-factor authentication will be strictly enforced for all environments.

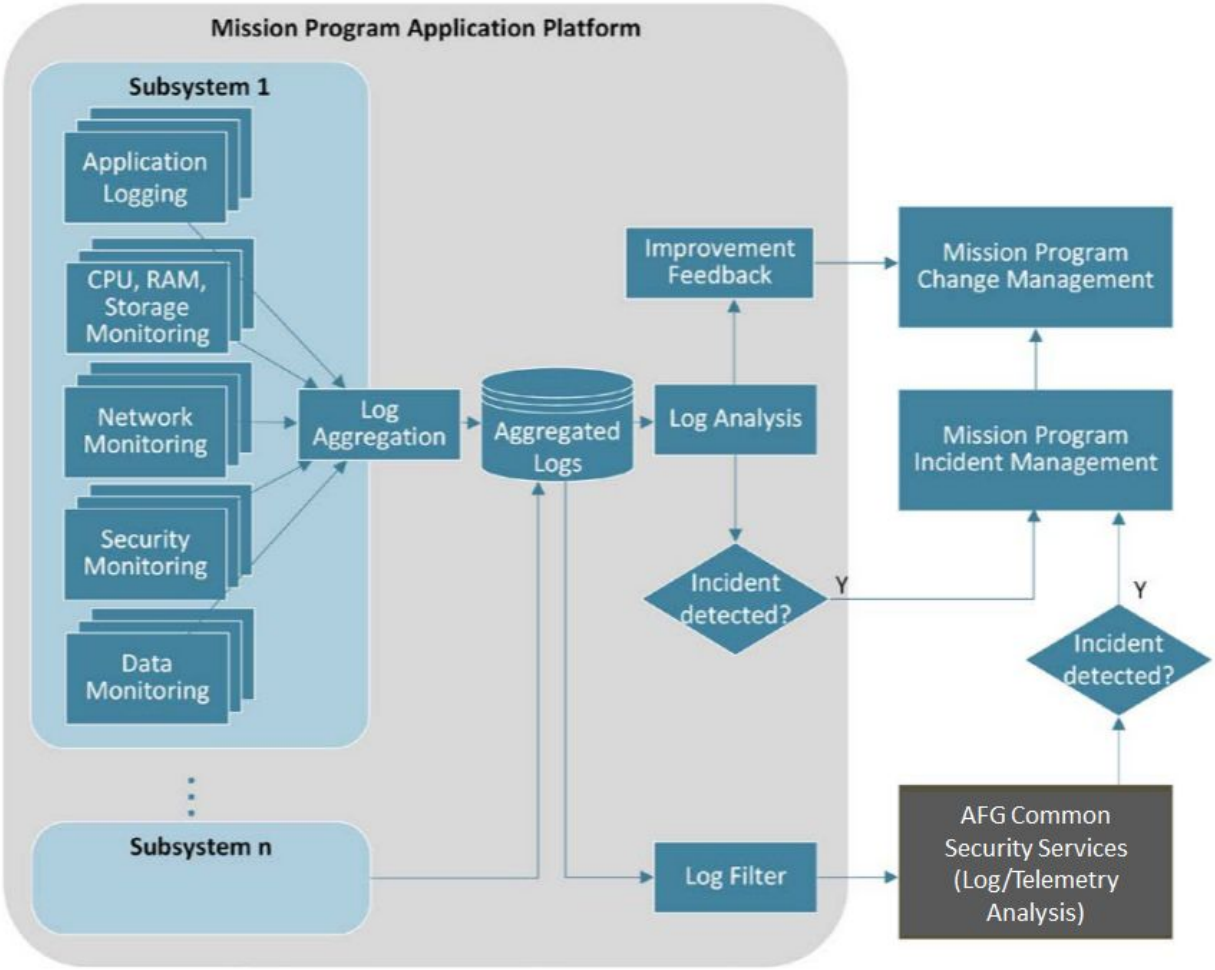


(Source: https://www.mvine.com/mv-tech-stack.html.)

Day 2 Operations - Dashboards

Observability is a key principle of DevSecOps. Every part of the system is instrumented for continuous telemetry. The figure below shows how application, systems, networking, and security metrics from each mission subsystem is aggregated into mission-specific databases for analysis at a mission-level or at a higher tier of management. Monitoring and alerting capabilities are built into the platform. Mission programs can set up their own monitoring and alerting rules as well as what actions are to be taken when an alert is detected. Security metrics are connected to the Intuitus Security Operations Center (SOC) and the capability for continuous diagnostics and automated mitigation. Network and system metrics are connected into a Network Operations Center (NOC) run by the AFG Compute team. Application-specific dashboards utilize application logging and data monitoring facilities. All logging instrumentation is part of the SCSS.

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

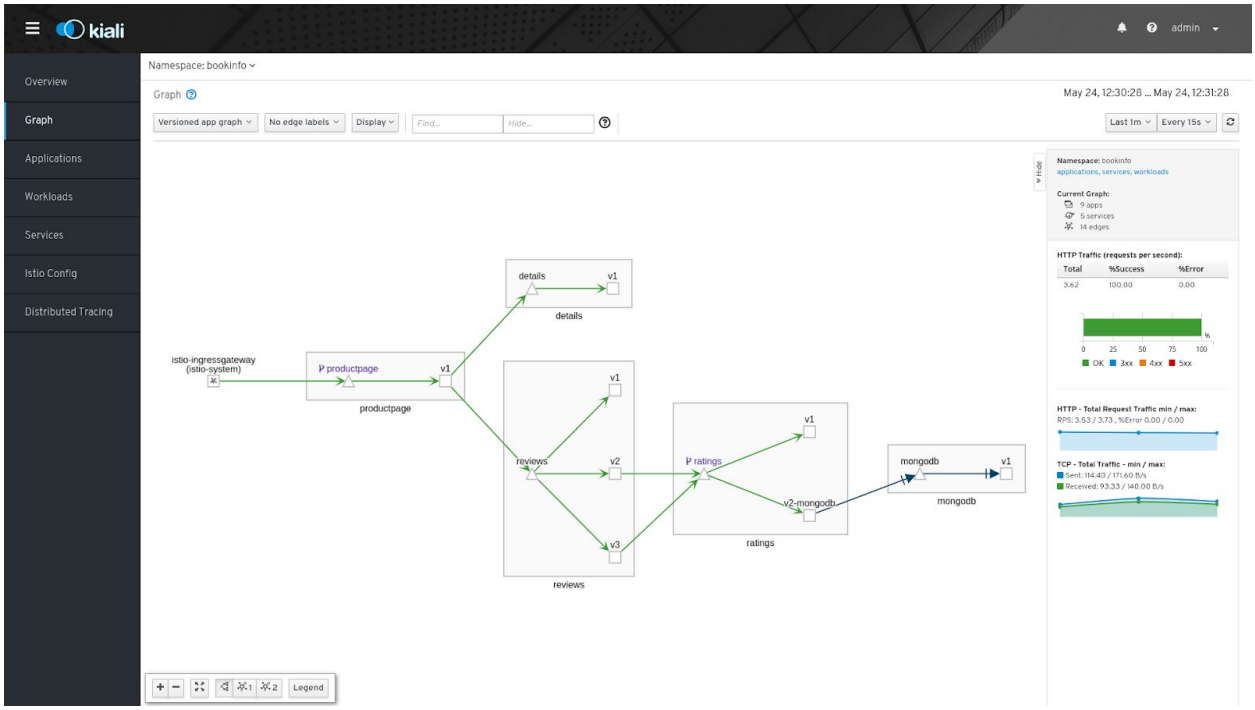


Application Operational Troubleshooting Dashboards

Troubleshooting distributed systems has traditionally been a challenge. Fortunately, the Kubernetes ecosystem is quite mature, with many open source tools such as Kiali and Jaeger.

According to http://kiali.io, Kiali is a management console for Istio-based service meshes. It provides dashboards and observability to let you operate your mesh with robust configuration and validation capabilities. It displays the structure of the service mesh by inferring traffic topology and displays the health of the mesh. Kiali provides detailed metrics, powerful validation, Grafana access, and strong integration for distributed tracing with Jaeger.

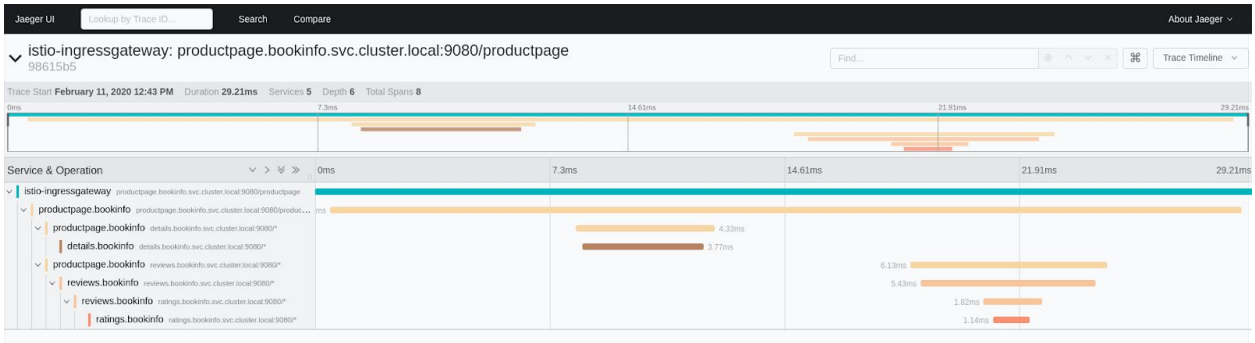
Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.



Jaeger is a distributed tracing system for monitoring and troubleshooting microservices-based distributed systems, including:

- Distributed context propagation
- Distributed transaction monitoring
- Root cause analysis
- Service dependency analysis
- Performance / latency optimization

When something is not working as expected in any deployment environment including production, Jaeger is the tool that can help perform end-to-end tracing. The following is an illustration of Jaeger tracing in Kiali:



Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

Application-specific Dashboards

Applications can have their own domain-specific dashboards. For example, the AFG Power team has its own dashboard consisting of one-line diagrams and digital twins, among other things. These are out of scope of this document, which covers common infrastructure for all applications.

Infrastructure Operations Dashboards

The AFG Compute team's systems operators want to make sure that as an application scales, it has all the necessary processing power, memory, networking capacity, and storage that it needs. Tools such as Prometheus and Grafana, integrated through the central logging, monitoring, and alerting system through the SCSS and Fluentd, can help produce dashboards like the following, which track key systems metrics, generate alerts and even take automatic action when certain thresholds are met or unmet as the case may be.

Cybersecurity Dashboards

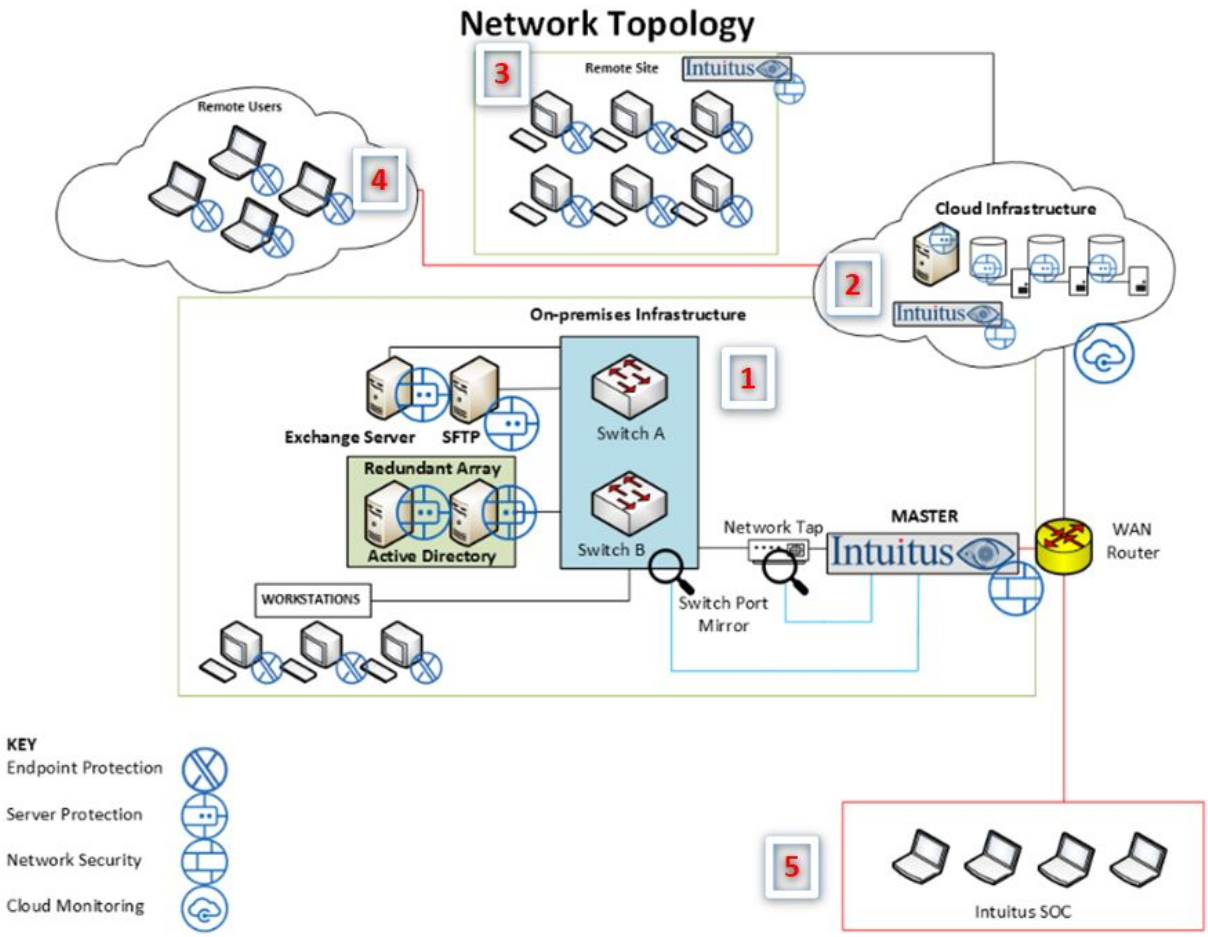
The Intuitus Security Management Appliance (SMA) in physical or virtual form can control access to and monitor networks and devices. The figure below shows a typical corporate network with the following components:

1. In a corporate office headquarters setting with its own on-premise networking equipment, an Intuitus SMA can be installed as a "master" device, able to collect and coordinate

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

command and control with other Intuitus SMAs in the corporation's other networks, described below.

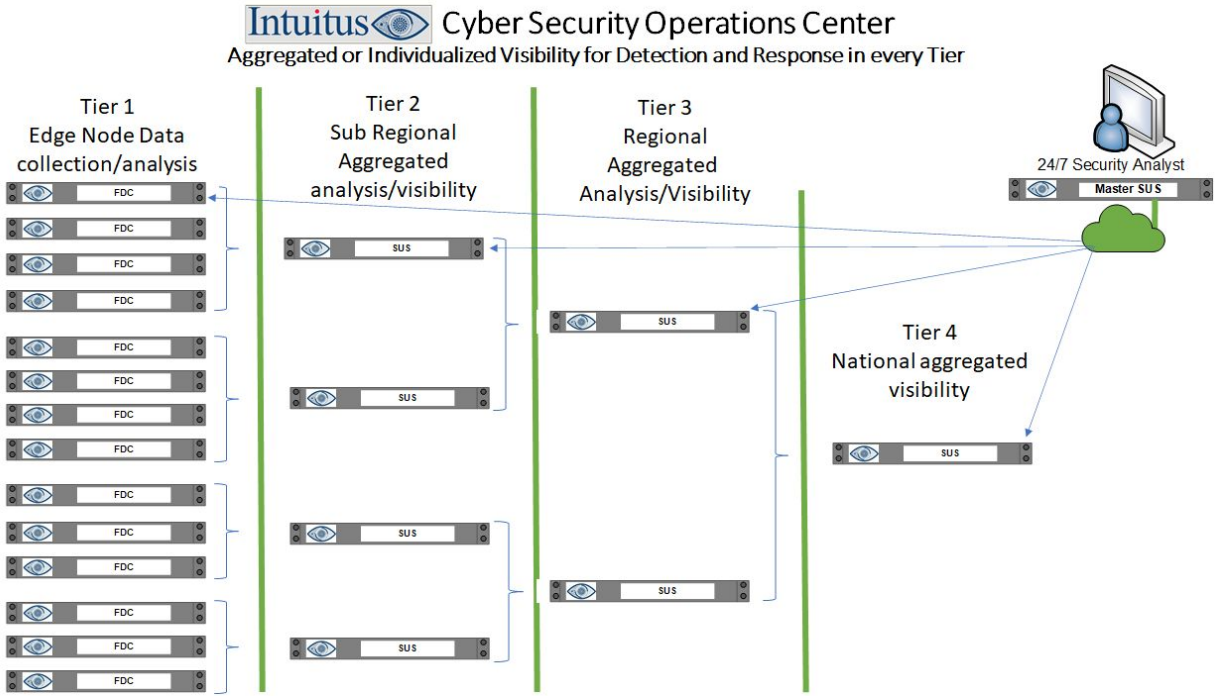
2. Where the corporation has deployed one or more applications in the cloud, a virtual SMA is attached to the cloud network.
3. On a remote site, a physical SMA does command and control for the local network there.
4. Remote users connecting to the corporation from various remote sites can be channeled through the cloud SMA.
5. Highly trained Intuitus cyber defense analysts are monitoring all network activity with the help of sophisticated software running on the SMAs as well as in the Security Operations Center (SOC).



Various dashboards are available to the cyber defense analysts. One of them is depicted below.

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

Intuitus SMAs can be tiered to local, district, regional, state, national, and international levels as needed to coordinate global or local networks, as illustrated in the following picture.



Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

Conclusion

The grant of authority to operate is based on provably secure processes being used in product development and delivery. The U.S. DoD has gone to great lengths to create an open and public template for any organization to create its own software factories using a veritable selection of open source and commercial tools without fear of vendor lock-in. Very few enterprises to date can claim to have a repeatable process with built-in provable security for ongoing continuous operation. Any enterprise looking to strengthen or transform its product development processes would be wise to consider using this template, while recognizing that a cultural transformation, not tools, is the key to success.

The goal of DBM is frictionless commerce for traditional enterprises looking to unshackle themselves and adapt the methodologies of purely digital enterprises. AFG's use of the DevSecOps approach to implement Smart Grids is a key enabler to frictionless delivery and our participation in DBM. Intuitus, AFG's core partner for cybersecurity and cyber physical systems security, oversees the DevSecOps implementation. We are happy to share our knowledge to help promote the growth and success of the DBM. Please feel free to contact the author to learn more.

References

https://software.af.mil/dsop/documents/ - this is a great repository of DoD DevSecOps documents, some of which are listed below.

https://dodcio.defense.gov/Portals/0/Documents/DoD%20Enterprise%20DevSecOps%20Reference%20Design%20v1.0_Public%20Release.pdf?ver=2019-09-26-115824-583

https://software.af.mil/wp-content/uploads/2020/09/DoD-Enterprise-DevSecOps-Initiative-Introduction-v5.5.pptx

https://software.af.mil/wp-content/uploads/2019/11/DoD-Enterprise-DevSecOps-Initiative-Hardening-Container-Document-v2.3.docx

https://software.af.mil/wp-content/uploads/2019/11/DoD-Enterprise-DevSecOps-Initiative-Moving-to-Microservices-Document-v1.3.docx

https://software.af.mil/wp-content/uploads/2019/11/DoD-Enterprise-DevSecOps-Initiative-Pipeline-Description-Document-v1.2.docx

https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/Contracting%20Considerations%20for%20Agile%20Solutions%20v1.0.pdf

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.

https://owasp.org/www-chapter-belgium/assets/2019/2019-02-20/Zero-to-DevSecOps-OWASP-Meetup-02-19-19.pdf

https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-137A.pdf

https://martinfowler.com/bliki/StranglerFigApplication.html

Portions Copyright © by Saro Saravanan and Intuitus Corp; others copyright their respective owners as per citations.