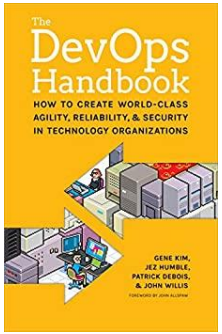
Red Hat
Global Transformation Office

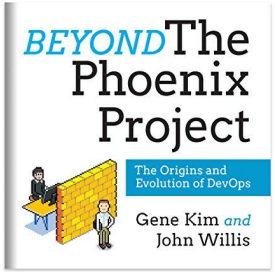
The Blurred and Broken Lines of Defense

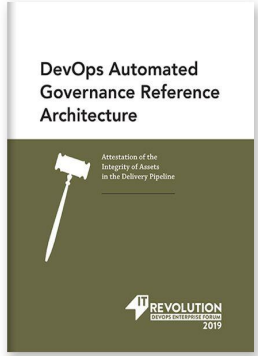
An Overview

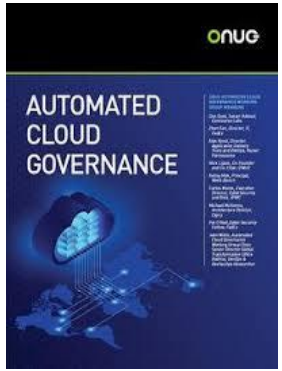
John WIllis
Global Transformation Office
@botchagalupe

@botchagalupe

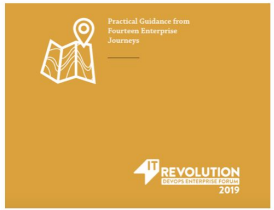


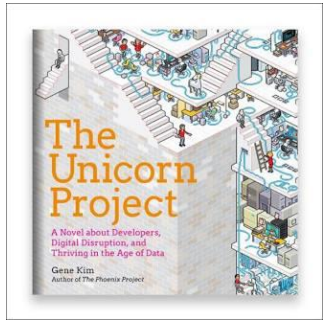






Four Frameworks for Portfolio Management





docker

CANONICAL

DELL

DEVOPS ENTERPRISE SUMMIT

KeyBanc Capital Markets

ExxonMobil

CHEF

DEVOPS DAYS

Red Hat

“What would DevSecOps be like if DevOps never existed?”

Blog

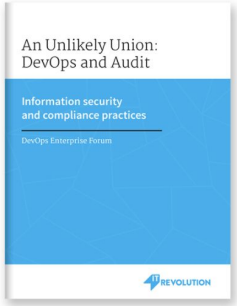
What is DevSecOps?

June 01, 2015 / Shannon Lietz

The purpose and intent of **DevSecOps** is to build on the mindset that ***"everyone is responsible for security"*** with the goal of safely distributing security decisions at speed and scale to those who hold the highest level of context without sacrificing the safety required.

Today, these same factors have led traditional security leadership to argue hard for a seat at the Executive table. And while having a seat at the table has increased the effectiveness of security decisions, it's since caused friction and a significant slow down in business outcomes because of a scarce supply of security skill sets to embed in the value creation process. Without enough people, the desired speed by business operators cannot be achieved and a change in how security value is contributed becomes necessary or risks to increase.

“Everyone is Responsible for Security”



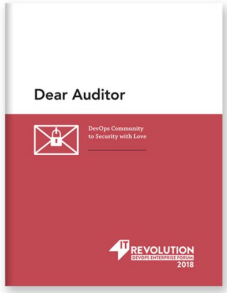
AN UNLIKELY UNION: DEVOPS AND AUDIT

October 1, 2015

DEVOPS AUTOMATED GOVERNANCE REFERENCE ARCHITECTURE

September 17, 2019

2015 2018 2019



DEAR AUDITOR

August 27, 2018

EXPLORE

Focusing on the DevOps Pipeline

Creating Better Pipelines

So how do we design, measure, and improve our pipelines to avoid the above?

Pipeline Design

At Capital One, we design pipelines using the concept of the "16 Gates". These are our guiding design principles and they are:

- Source code version control
- Optimum branching strategy
- Static analysis
- >80% code coverage
- Vulnerability scan
- Open source scan
- Artifact version control
- Auto provisioning
- Immutable servers
- Integration testing
- Performance testing
- Build deploy testing automated for every commit
- Automated rollback
- Automated change order
- Zero downtime release
- Feature toggle

These gates are used to understand each and every product's progress through the DevOps process.

Industry Working Groups

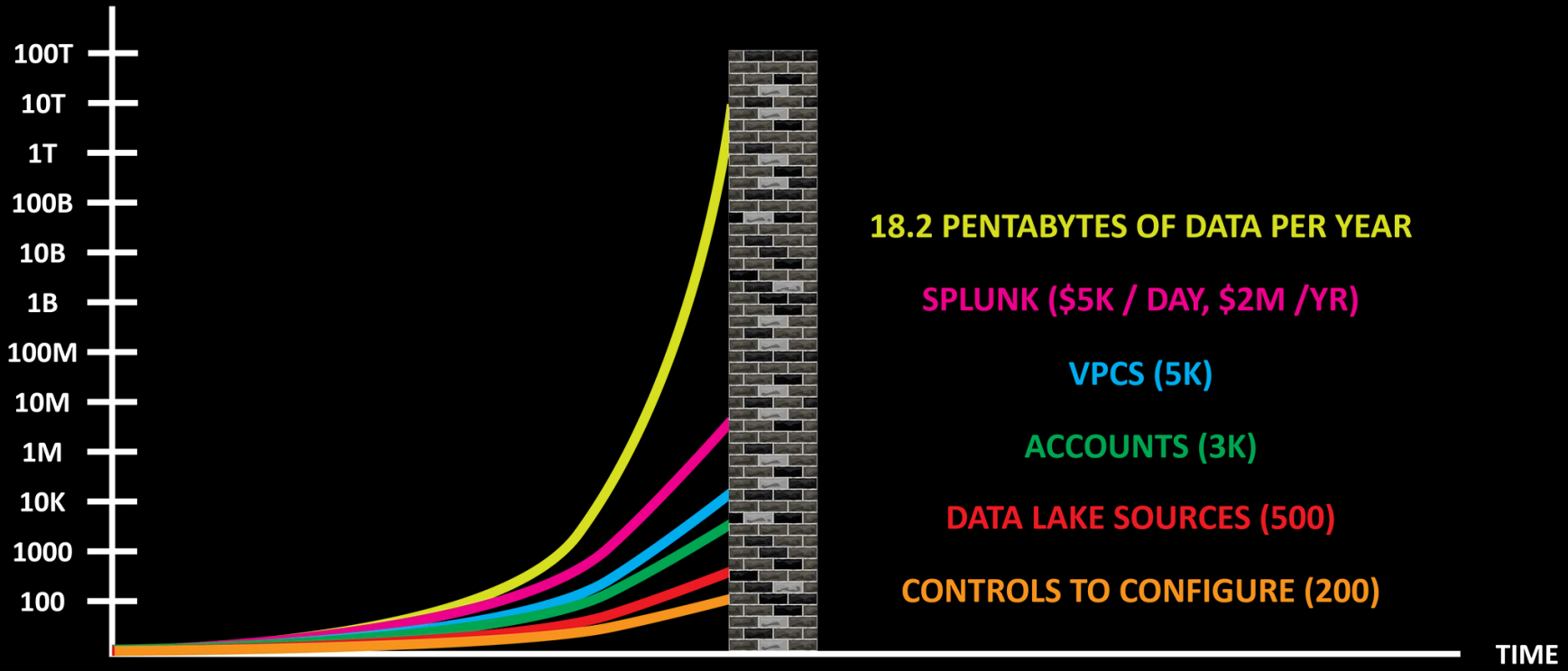
- Automated Governance (Risk)
- Automated Cloud Governance (Defense)

DEVOPS AUTOMATED GOVERNANCE REFERENCE ARCHITECTURE

September 17, 2019

Red Hat

Minimum Viable Security Posture



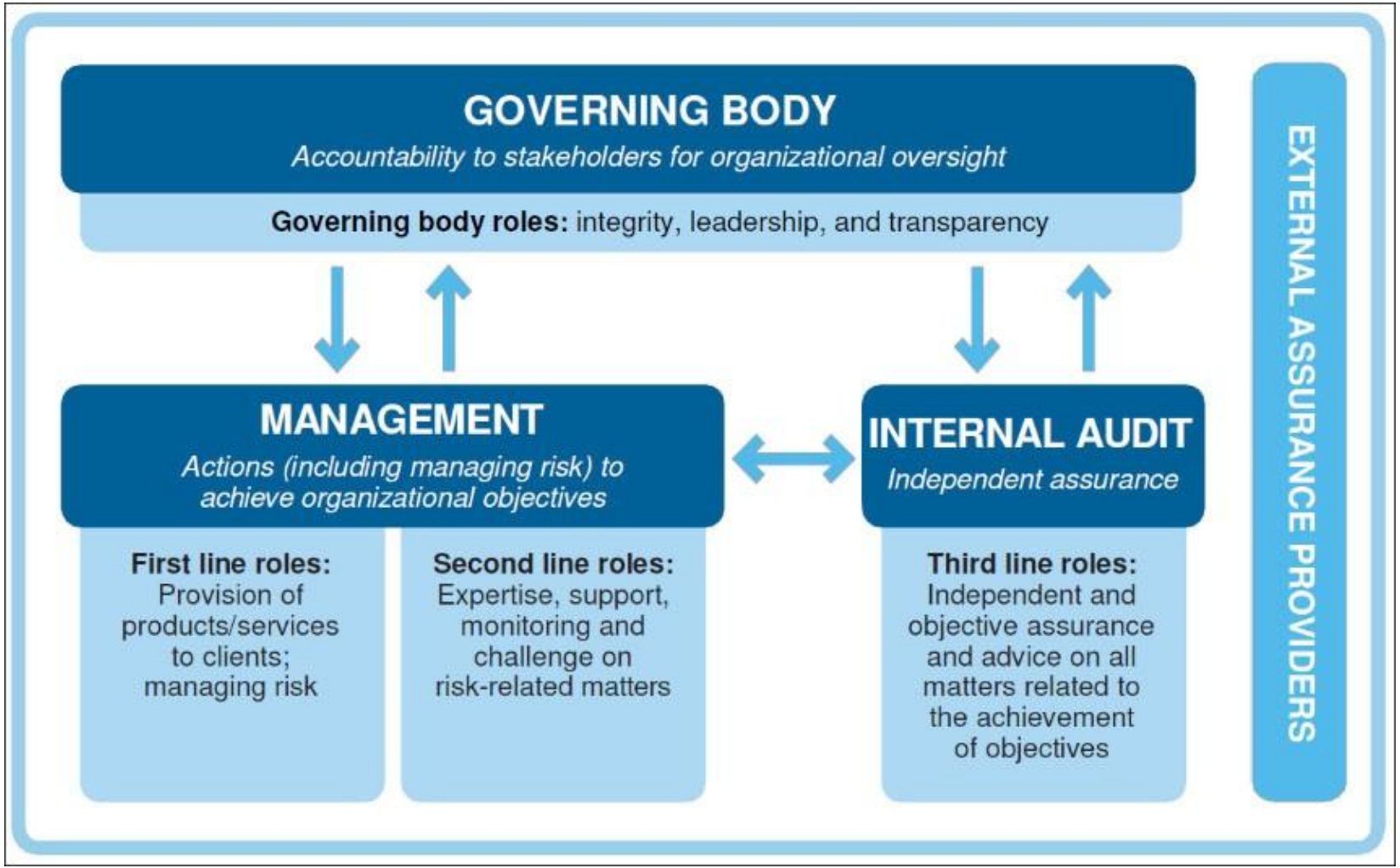
ONUG

“Traditional Security Models are Anti-Patterns”

The Three Lines of Defense Model

Source: The Institute of Internal Auditors

“Any organization that designs a system will produce a design whose structure is a copy of the organization's communication structure”

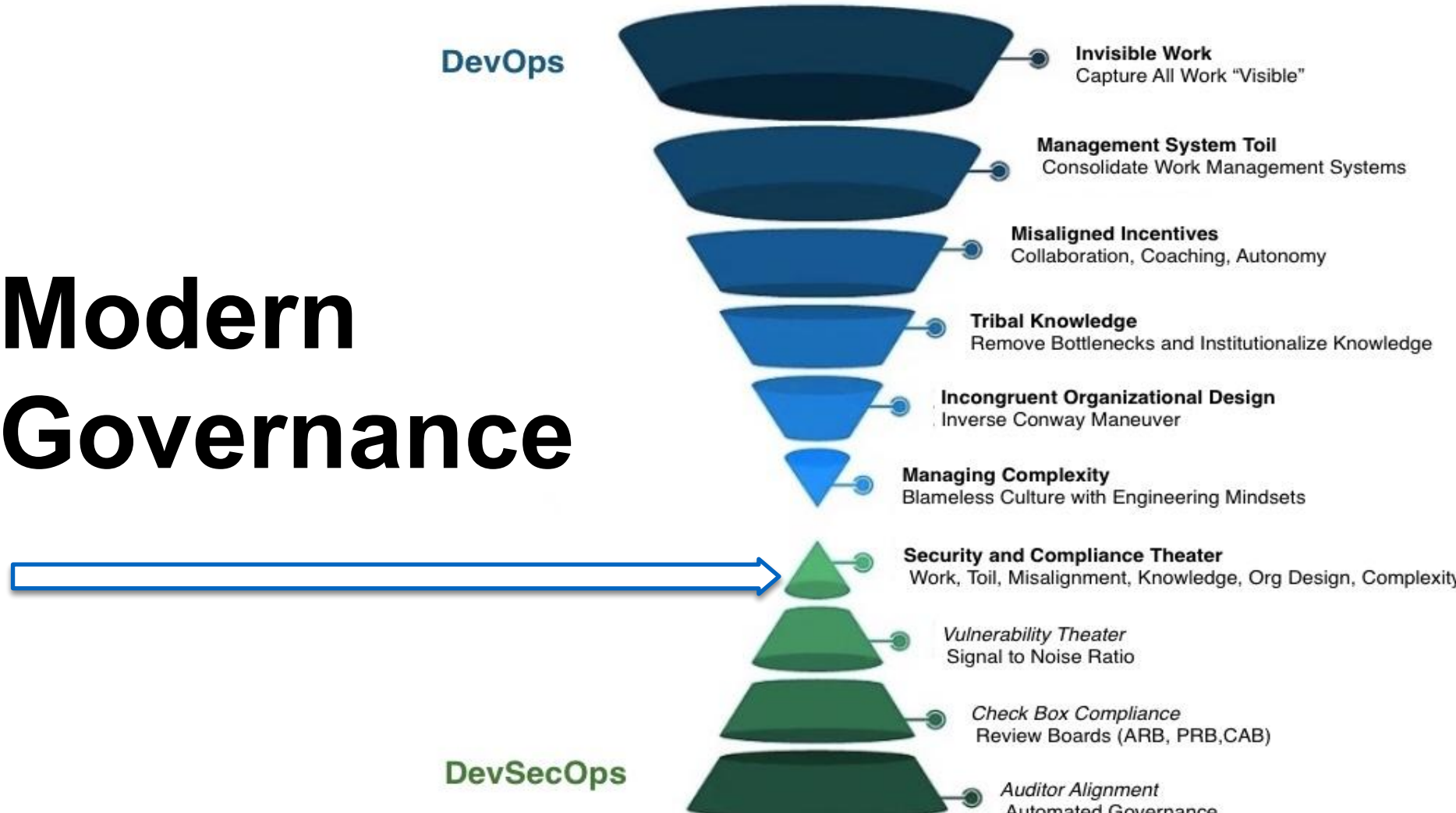


Institute of Internal Auditors 2020

FS ISAC Related - Banking Regulatory 2020 Outlook

“Another common issue is having an essential capability that is missing or inadequate in one line of defense but present in a different line. For example, if the business (first LOD) does not have testing capabilities, the risk and/or compliance functions (second LOD) might perform testing; however, if the second LOD’s testing is deemed inadequate, internal audit (third LOD) might perform additional testing.”

Modern Governance



Modern Governance

- **Modern Risk**
  - Toil and Efficacy related to Risk
  - Governance and Compliance
  - Attestation and Enforcement
- **Modern Defense**
  - Toil and Efficacy related to Cyber Defense
  - MIRTE/NIST/FedRAMP
  - Intelligence/Cyber Data Lake
- **Modern Trust**
  - Toil and Efficacy related to Identity
  - Platform-Agnostic Authentication, Cryptographic Identities
  - Zero Trust Models/Secure Production Identity Frameworks (SPIFFE)

DevOps Automated Governance (Risk)

Verifiable Data Audit

- How do I prove that I'm safe?
- How do I demonstrate that I'm secure?
- How do I know that I can make those statements in a way that's trustful, that I can actually have evidence that stands behind it?

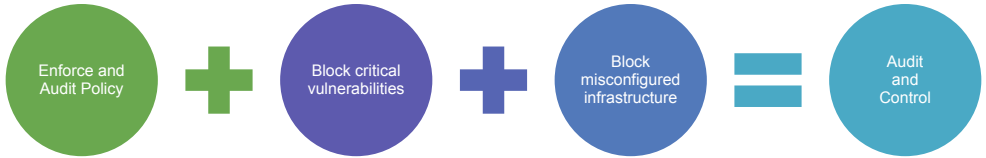
Automated Governance (What)

Move from an implicit trust based model for controls to an explicit, proof based model.

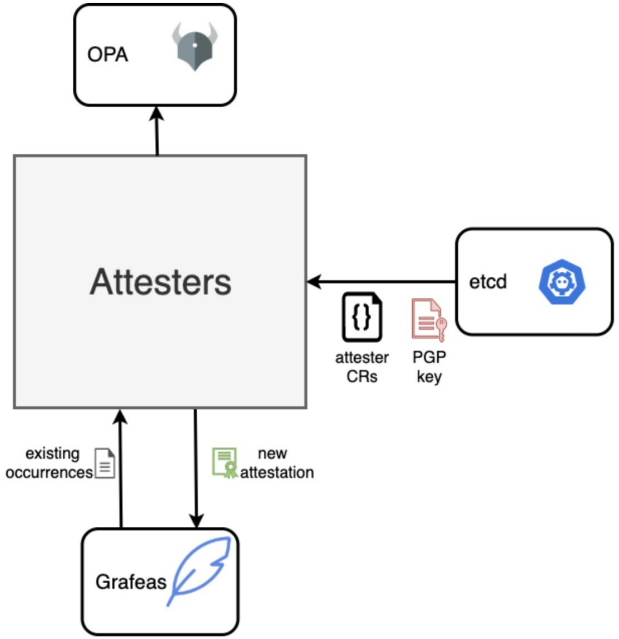
Automated Governance (How)

Changing Subjective attestation into Objective attestation

Objective Evidence and Closed Feedback Loops

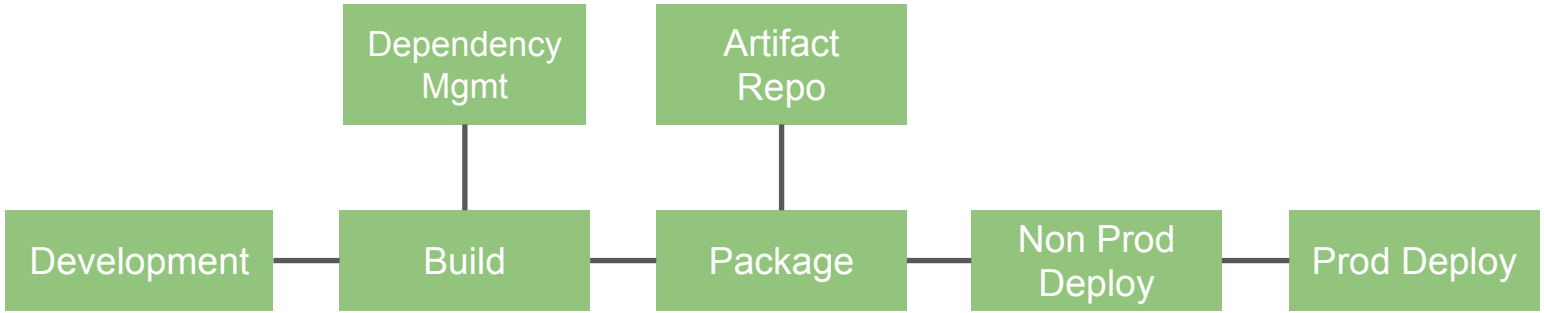


- Reduce Audit Time
- Increase Audit Efficacy
- Shorten Feedback Loops
- Local Authority
- Minimize Handoffs
- 90% of Controls are Manual
- Enable Trust



Red Hat

Devops automated Governance Reference Architecture



Common Control

1. Access Control
2. Audit Train/log
3. Everything source control
4. Usage policies

Common Actors

1. Auditor, Risk/Compliance Office
2. (system)
3. Tools Admin

Development Stage

Attestation Name	Stage	Attestation	Source	Example
Code Quality	Development	Numeric	SonarQube	<4
Information Leakage	Development	Pass/Fail	Custom	pass
Unit Test Coverage	Development	Percentage Code Coverage	SonarQube	80%
Unit Test Execution	Development	Pass/Fail	SonarQube	Pass
Change Size	Development	Pass/Fail	Jenkins	Pass
Cyclomatic Complexity	Development	Pass/Fail	SonarQube	Pass
Pull Request	Development	Number of Approvers	Source Control	
Branching Strategy	Development	Pass/Fail	SonarQube	Pass
Clean Dependencies	Development	Validation	Nexus	validated

Build Stage

Attestation Name	Stage	Attestation	Source	Example
Build	Build	ID	Source Control	2.0.3-16-98092ba
Build Performance	Build	Verification		verified
Build Version	Build	Version	Source Control	98092ba
Build Conconfiguration	Build	Config Identfication	Source Control	
Linting	Build	Pass/Fail	SonarQube	Pass
SAST Scan	Build	Validation	Sonarqube	Invalid

Package Stage

Attestation Name	Stage	Attestation	Source	Example
Artifact Versioning	Package	Pass/Fail	Nexus	Pass
Package Metadata	Package	Pass/Fail	Nexus	Pass
Code Signing	Package	Validation	Cryptographic Hash	validated
Container Scan	Package	Validation	Openscap	validated

PreProd Stage

Attestation Name	Stage	Attestation	Source	Example
Trusted Packages	Pre-Prod	Validation	Openscap	validated
Approved Configuration	Pre-Prod	Validation	Ansible	validated
Threat Monitoring	Pre-Prod	Validation	Tanium	validated
Autom Alert Tooling	Pre-Prod	Validation	Dynatrace	validated
Deployment Strategy	Pre-Prod	Validation	Ansible	validated

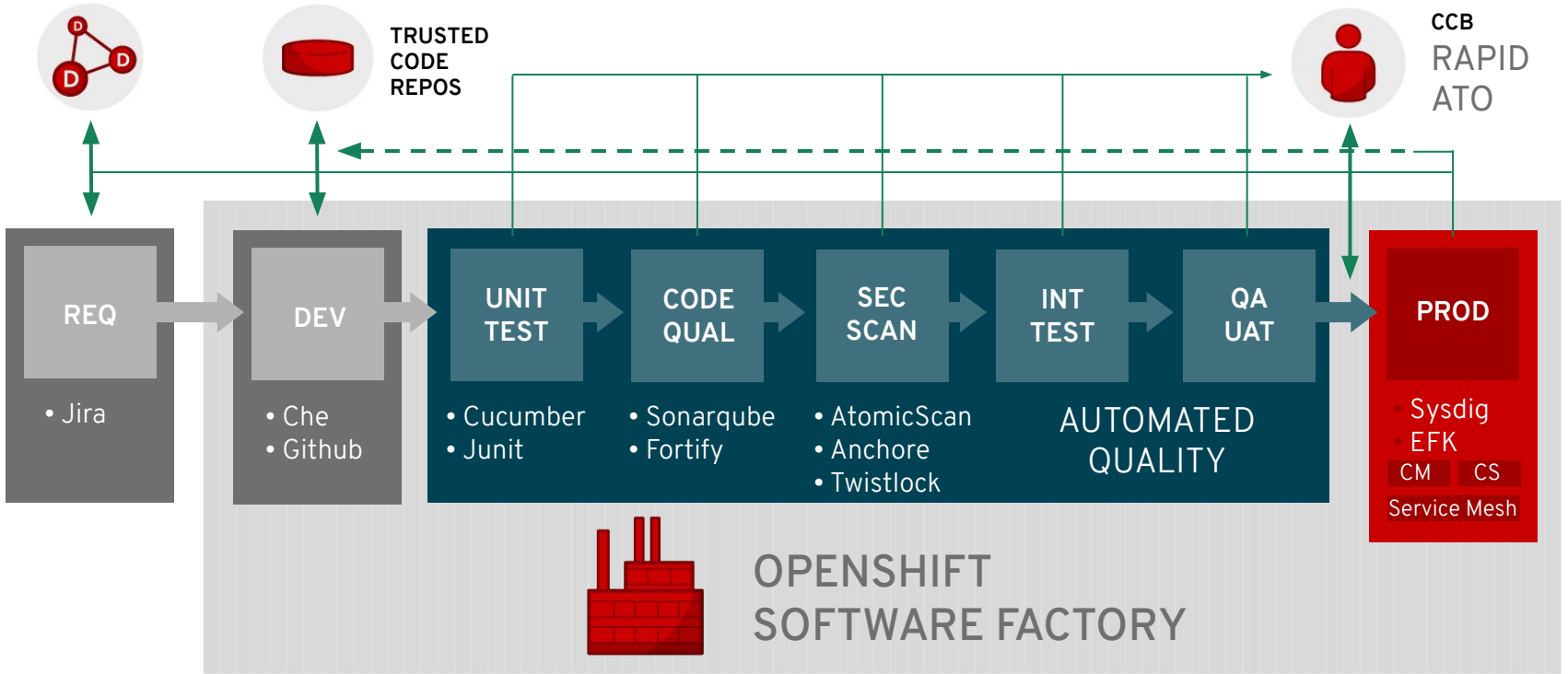
Risk as Code

- Human Readable (YAML)
- Machine Interpreted
- Version Controlled
- Models Attestations and Enforcement

```yaml
apiVersion: pac/v1
kind: Policy
metadata:
  name: azn_services
  labels:
    mnemonic: azn
    app: azn_web
    requirements:
    #Versioning
    pipeline:
      versioning:
        verify: True
        pattern: `^(\d+)\.(\d+)\.(\d+)-(\d+)\-([A-Faa-f0-9]+)$`
    #Build server Verification
    build_server:
      verify: True
    #Artifact Server Verification
    artifact_server:
       verify: True
    #Artifact Verification
    artifact_hash:
       verify: True
    #Unit Testing
    unit_test_coverage:
       verify: True
       percentage_coverage: at_%_coverage
    #Git Pull Reguest
    pull_request_approval:
       verify: True
       pull_request_approver_count:
```

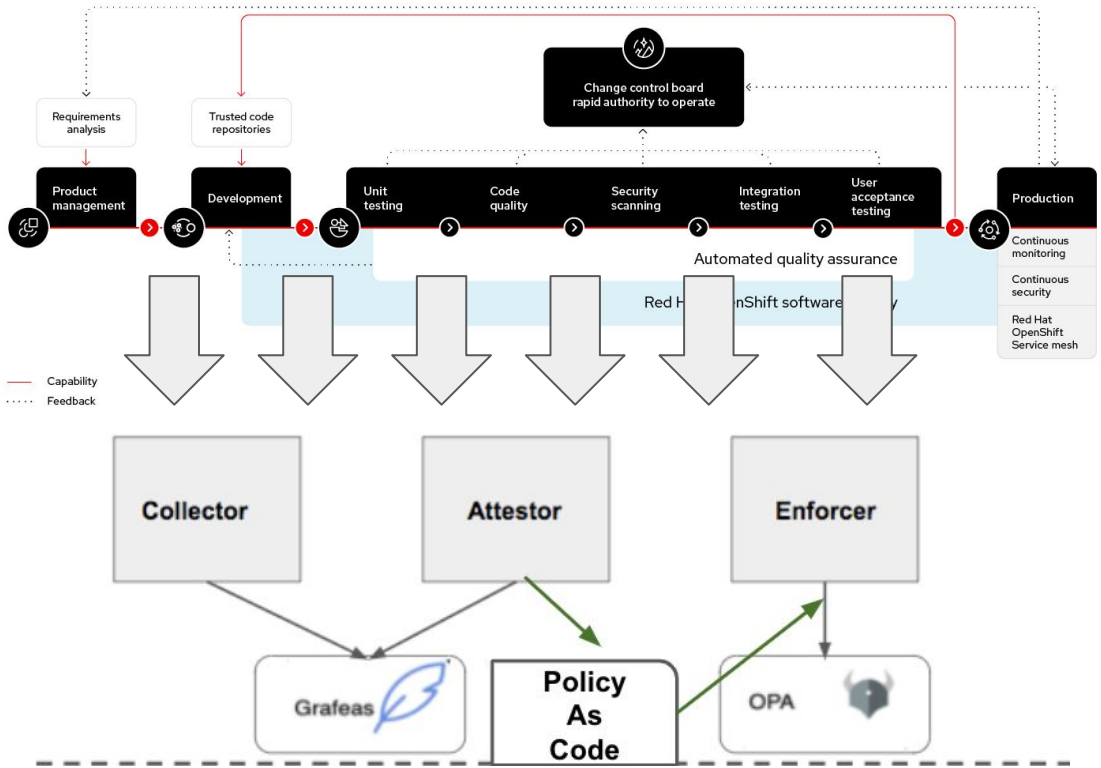
The Trusted Software Supply Chain

POWERED BY Red Hat OpenShift



Red Hat OpenShift

Trusted software supply chain



Idea To Production

Red Hat's trusted software supply chain (TSSC) automates all aspects of trust. It enforces critical processes required for software development & deployment..

Red Hat

tssc 0.15.0

```
pip install tssc
```

Latest version

Released: Dec 17, 2020

Trusted Software Supply Chain (TSSC) python library.

Navigation

Project description

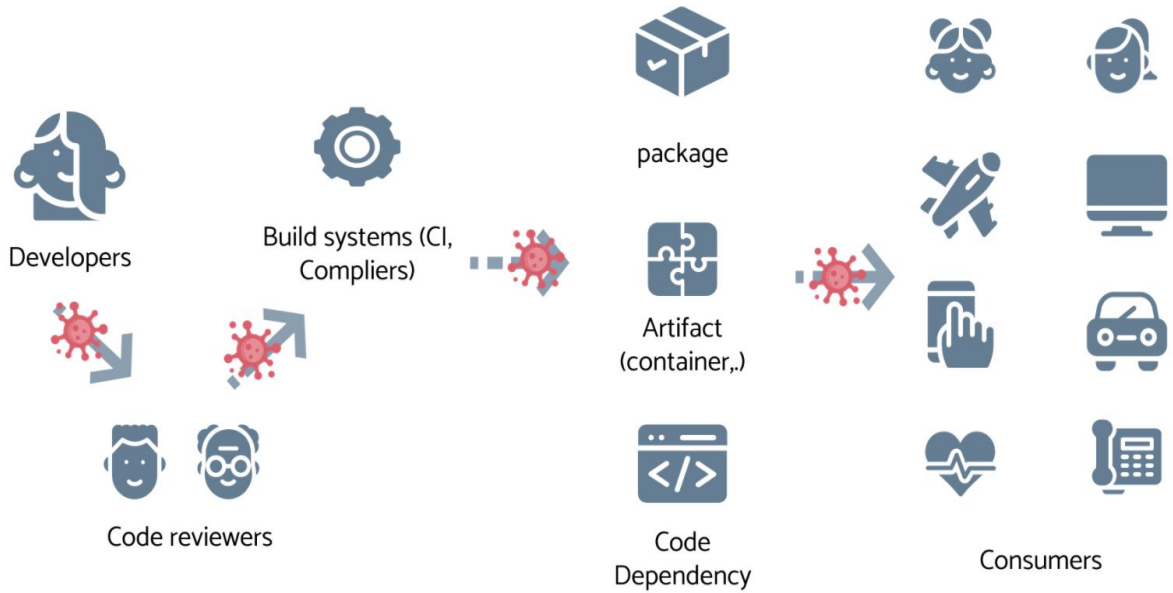
Release history

Download files

Project description

tssc-python-package

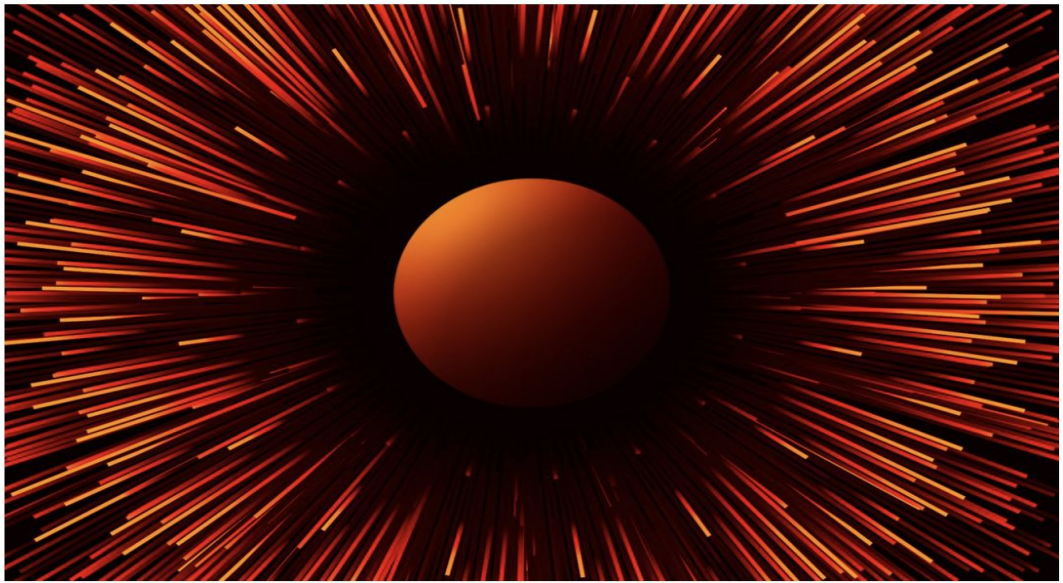
- Replay / freeze attacks
- Compromised keys
- SSO Compromise
- Malicious hashes
- Compromise of build systems
- Easy reconnaissance (open configuration)



Rekor seeks to combine **provenance**, **integrity**, and **discoverability** to create a transparent and auditable software supply chain:

SUNSPOT: An Implant in the Build Process

January 11, 2021 CrowdStrike Intelligence Team Research & Threat Intel



In December 2020, the industry was rocked by the disclosure of a complex supply chain attack against SolarWinds, Inc., a leading provider of network performance monitoring tools used by organizations of all sizes across the globe. CrowdStrike and another firm have been supporting SolarWinds in its

CATEGORIES

ENDPOINT & CLOUD SECURITY (207)
ENGINEERING & TECH (25)
EXECUTIVE VIEWPOINT (107)
FROM THE FRONT LINES (120)
IDENTITY PROTECTION (11)
PEOPLE & CULTURE (18)
REMOTE WORKPLACE (18)
RESEARCH & THREAT INTEL (136)
TECH CENTER (107)

CONNECT WITH US

Tactic	ID	Technique	Attestation	Attestation Source	Observation
Reconnaissance	T1592.002	Gather Victim Host Information	Custom	TSSC/PyPi/Pipeline as Code	StellarParticle had an understanding of the Orion build chain before SUNSPOT was developed to tamper with it.
Resource Development	T1587.001	Develop Capabilities – Malware	Custom	TSSC/PyPi/Pipeline as Code	SUNSPOT was weaponized to specifically target the Orion build to replace one source code file and include the SUNBURST backdoor.
Defense Evasion	T1140	Deobfuscate/Decode Information	Configuration	Inspect	The configuration in SUNSPOT is encrypted using AES128-CBC. It contains the replacement source code, the targeted Visual Studio solution file name, and targeted source code file paths relative to the solution directory.
Defense Evasion	T1027	Obfuscated Files or Information	Configuration	OSCAP, Rekor,Trillion	The log file SUNSPOT writes is encrypted using RC4.
Defense Evasion	T1480	Execution Guardrails	Code Signing	Cryptographic Hash	The replacement of source code is done only if the MD5 checksums of both the original source code file and backdoored replacement source code match hardcoded values.
Defense Evasion	T1480	Execution Guardrails	Image Scanning	OSCAP	The replacement of source code is done only if the MD5 checksums of both the original source code file and backdoored replacement source code match hardcoded values.
Defense Evasion	T1036	Masquerading	Custom	Rekor,Trillion	SUNSPOT masquerades as a legitimate Windows Binary, and writes its logs in a fake VMWare log file.

MITRE ATT&CK Framework Analysis from “Crowdstrike Blog - SUNSPOT: An Implant in the Build Process

Thank you

jwillis@redhat.com
@botchagalupe


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat

Cloud Automated Governance (Defense)

Cloud Automated Governance

WORKING GROUP

DON DUET

concourse labs

RAJ BALASUBRAMANIAN

IBM

JONATHAN BITLE

KAISER PERMANENTE

PETER CAMPBELL

Cigna

YURI CANTOR

Goldman Sachs

ZHEN FAN

FedEx

LENNY GRINBERG

Pfizer

MARK TIERNEY

ONUG

PRIYA GNANASUNDAR

KAISER PERMANENTE

ALI ILOGLU

UBS

ANURAG JAIN

Goldman Sachs

ALEX KVYAT

KAISER PERMANENTE

CARLOS MATOS

JPMorgan Chase & Co.

PAUL MATUSIK

Raytheon Technologies

JOHN WILLIS

Red Hat

CHARLENE O'HANLON

PAT O'NEIL

FedEx

ANATOLIY PANASYUK

Microsoft

SMRITI TALWAR

IBM

MICHAEL WHEELER

Cigna

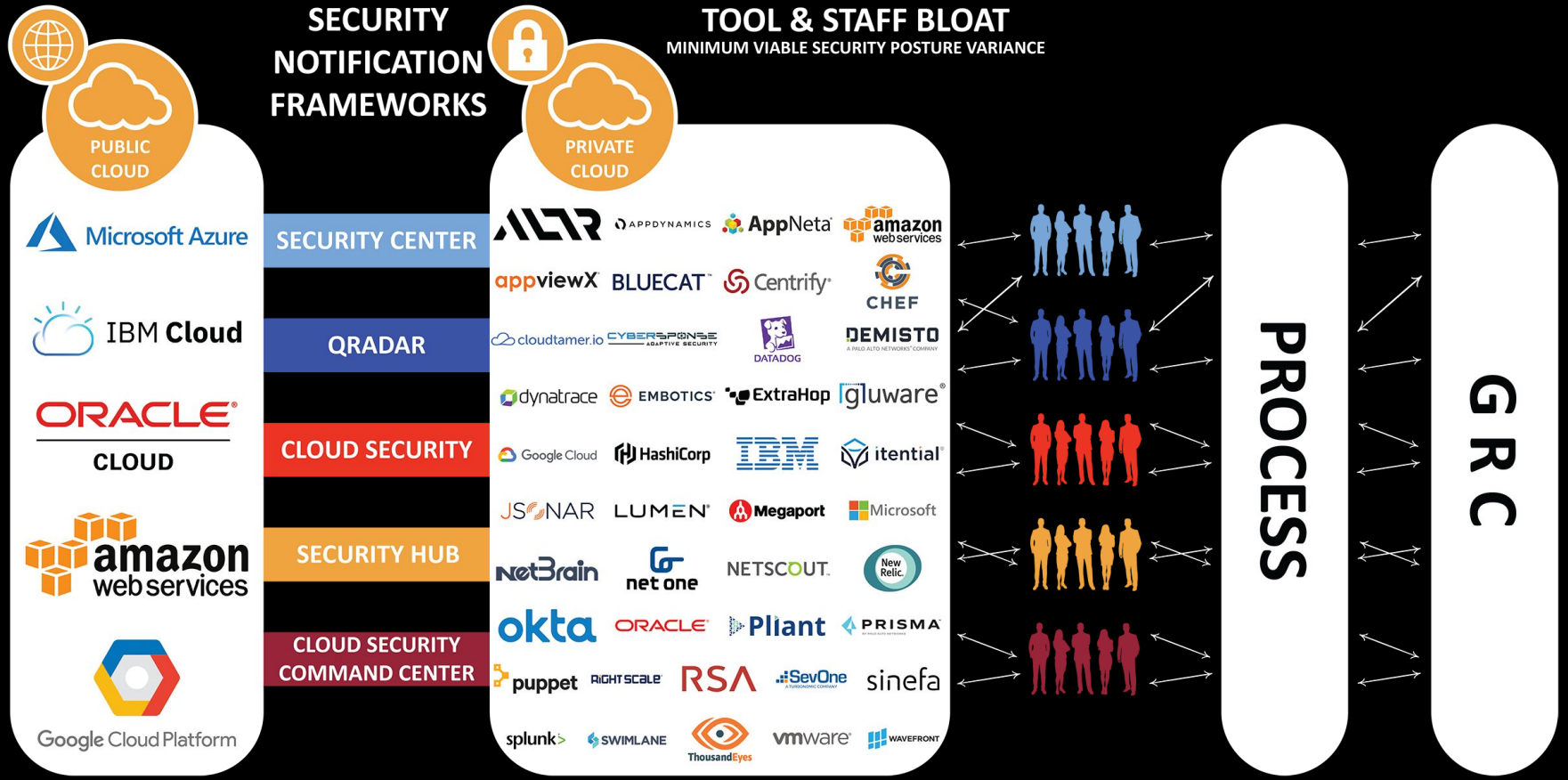
RANDY SHORE

cloudtamer.io

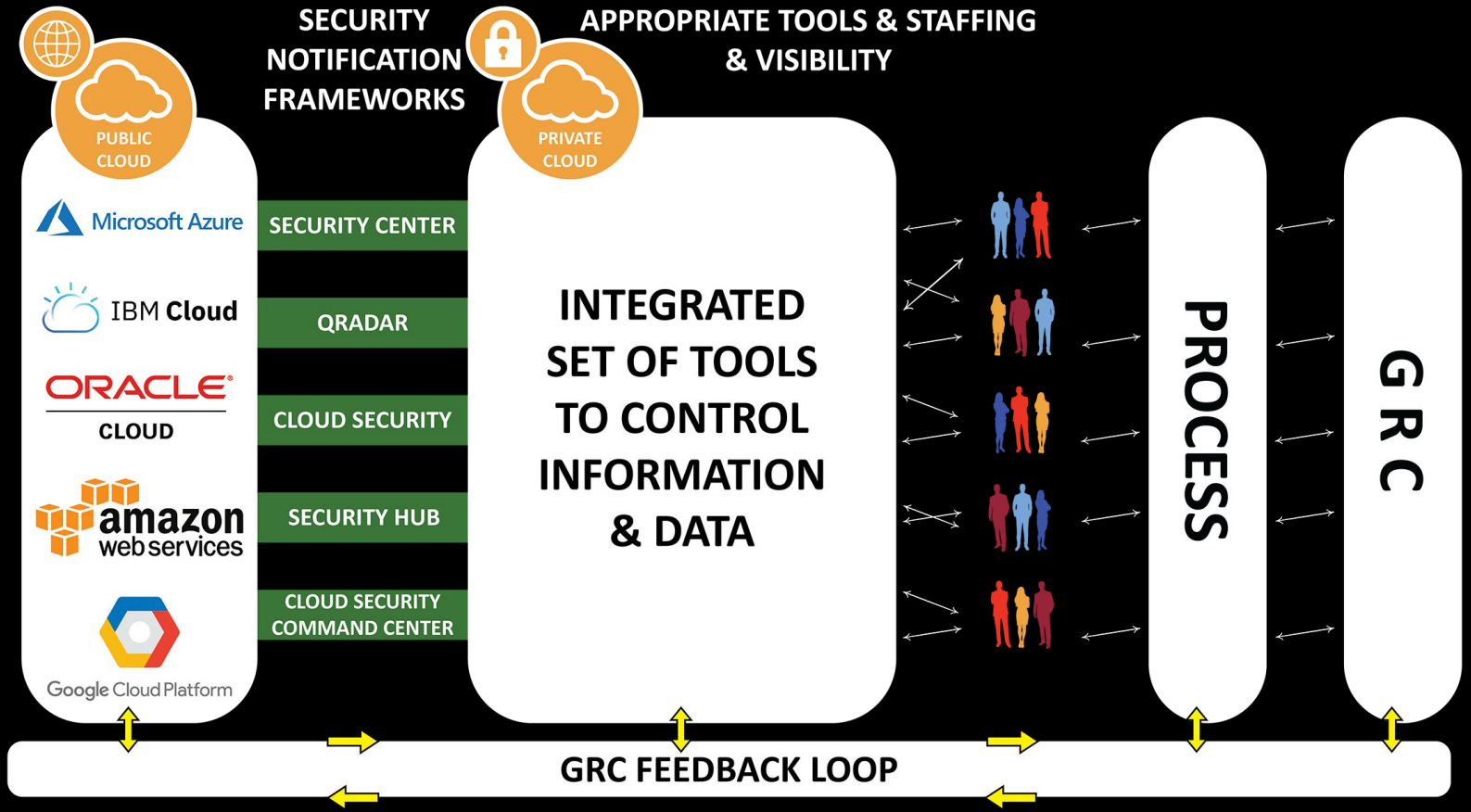
JOSEPH SPURRIER

cloudtamer.io

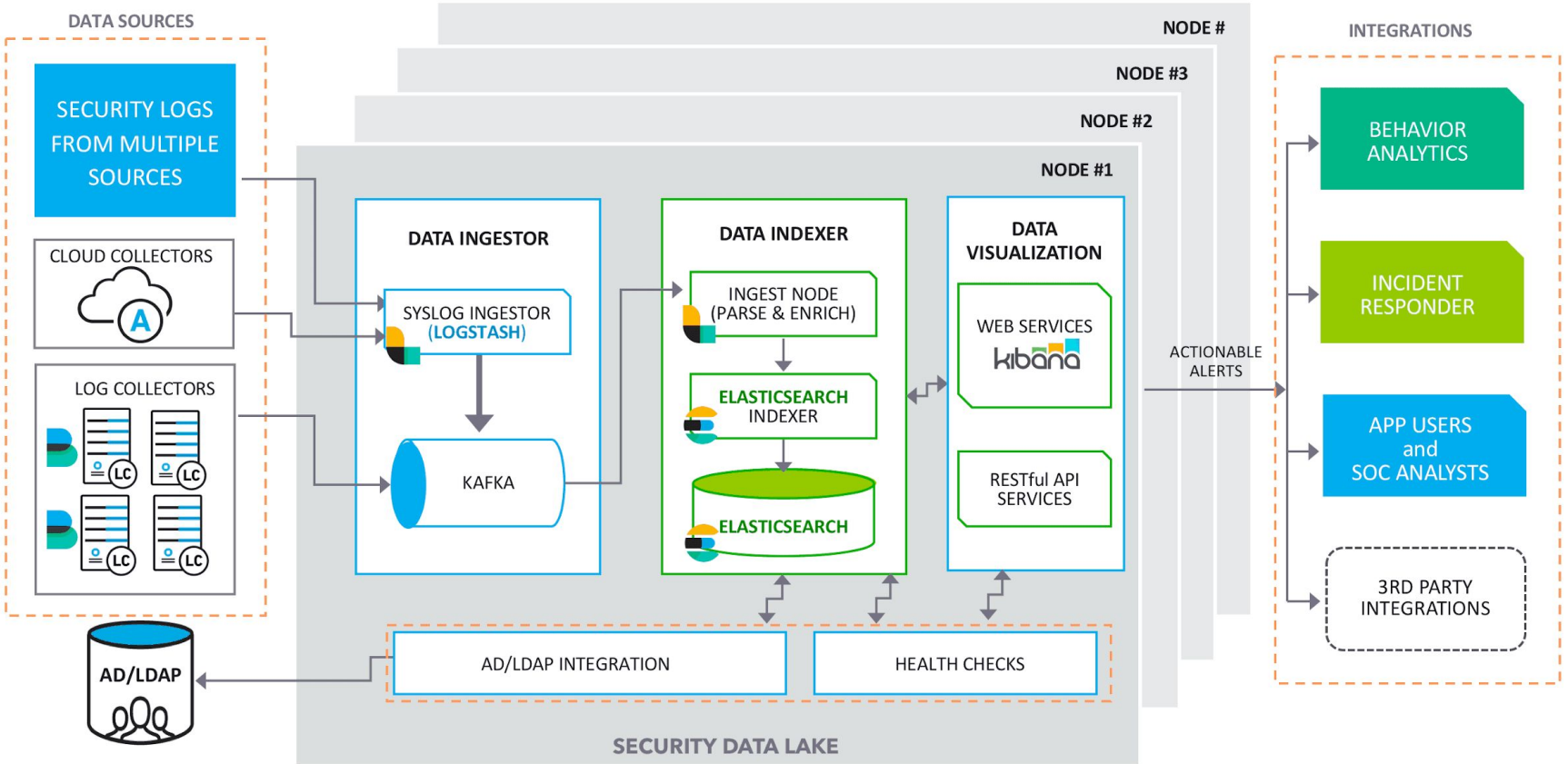
CLOUD CONTROL PROBLEM EMERGING



AUTOMATED CLOUD GOVERNANCE DESIRED OUTCOME



Security Data Lake



https://www.exabeam.com/siem/data-lakes/

Identity

Trust Opportunity

- Secure Production Identity Framework for Everyone (SPIFFE)
- Zero Trust Architecture
- Google’s BeyondCorp
- Secrets Management
- Meta Servers