# WIE SICHER IST EIGENTLICH „DEVOPS“? –
## PRODUKTIONS-DEPLOYMENTS OHNE VIER-AUGEN-PRINZIP?

Jens Borchers

Beratung für Informationsmanagement, Hamburg

ASQT 2020 – „SW Quality – looking forward & looking backward”

Bozen/virtuell - 11.11.2020

cat out

# WIE SICHER IST EIGENTLICH „DEVOPS“?

Version: 1.0 / 11.11.2020

Wie sicher ist eigentlich „DevOps“? – Produktions-Deployments ohne Vier-Augen-Prinzip?



Jens Borchers
Beratung für Informationsmanagement
Hamburg
jens@borchers-bfi.de
Jens.borchers@cat-out.com

# AGENDA

1|DevOps und Regulatorik Revisited

- Rückblick auf 2017 und Update
- DevOps-Nutzung heute

2|Bedrohungslage - „Zero Trust“

- Aktuelle Cybersecurity-Rahmenbedingungen
- Mögliche Angriffsvektoren für DevOps

3|Umsetzung von „Zero Trust“ für DevOps

- Zero Trust Basis-Prinzipien
- Identity und Access Management als zentrale Bestandteile
- DevSecOps – Automatisierte Compliance

4|Fazit

# 1 – DevOps und Regulatorik revisited

## Stand 2017 und heute

- Hauptdiskussion 2017: Ist DevOps in regulierten Umgebungen (insbesondere Finanzdienstleistern) legal?
  - Antwort 2017: Mit großer Wahrscheinlichkeit nicht!
  - Aber in vielen Banken werden agile und damit auch DevOps-Entwicklungen verbreiteter
  - Stand heute: Bisher kein offizielles Statement von BaFin zur Zulässigkeit, die „regulatorische Gefahr“ ist nicht verschwunden!
  - Aber: Erste Banken nutzen DevOps
- Die Empfehlungen von 2017 prinzipiell noch gültig
  - Aber die Bedrohungslage ist heute eine andere!
  - Cybersecurity ist in den Vordergrund gerückt


AGILITY MEETS REGULATION
Konfrontation oder Kooperation?

Jens Borchers
ASQT 2017 in Graz – 10.11.2017

sopra steria CONSULTING

EINSATZ VON AGILEN PROZESSEN BIS ZUM BETRIEB
FAZIT 1 – BEWERTUNG DER AKTUELLEN LAGE

- Der agile Entwicklungsteil ist aus Regulatorik-Sicht „ungefährlicher“,
  - Die Aufsicht macht keine Vorgaben, wie man Software entwickelt
  - Aber: Dokumentationsanforderungen gelten immer!
  - Und Nachvollziehbarkeit von der Fachanforderung bis zur Inbetriebnahme der entsprechenden Software-Komponenten
  - Fazit: Überlebenswichtige Kernsysteme mit klar definierten Anforderungen und geregelten Release-Zyklen weiter eher im Wasserfallmodell belassen.
- DevOps (in „Reinkultur“) bleibt vermutlich aus Sicht der Aufsicht „illegal“:
  - Klare Trennung von Zuständigkeiten
    - „Entwickler darf Betrieb nicht direkt mit Software versorgen“ – 4-Augen-Prinzip
  - Organisatorische Trennung von Entwicklung und Betrieb
    - Trend zur Zusammenlegung der „Change“- und „Run“-Organisationen wird eher kritisch gesehen
  - Die mit ITIL etablierten rigiden „Request for Change“-Prozesse lassen sich nur in Teilen automatisieren.

# 1 – DevOps at a Glance

## Grundlegende Konzepte

- Seit 2007 von Patrick Debois und  Andrew Shafer propagiert, 2009 durch John Allspaw and Paul Hammon bekannt gemacht in “10 Deploys a Day: Dev and Ops Cooperation at Flickr”
  - Organisation, Prozesse,  Werkzeugketten und vor allem “Kultur”
  - Hauptziel: Bessere Integration von Entwicklung und Betrieb
- Heutige Umsetzung
  - DevOps primär werkzeug-zentriert
    - Code Repositories
    - Automatische Qualitätssicherung und Testen
    - Continuous Integration/Continuous Deployment
    - Häufig in Verbindung mit Microservice-Architekturen
  - DevSecOps
    - Absicherung der Prozesse und Infrastrukturen


Grafik: informatik-aktuell.de

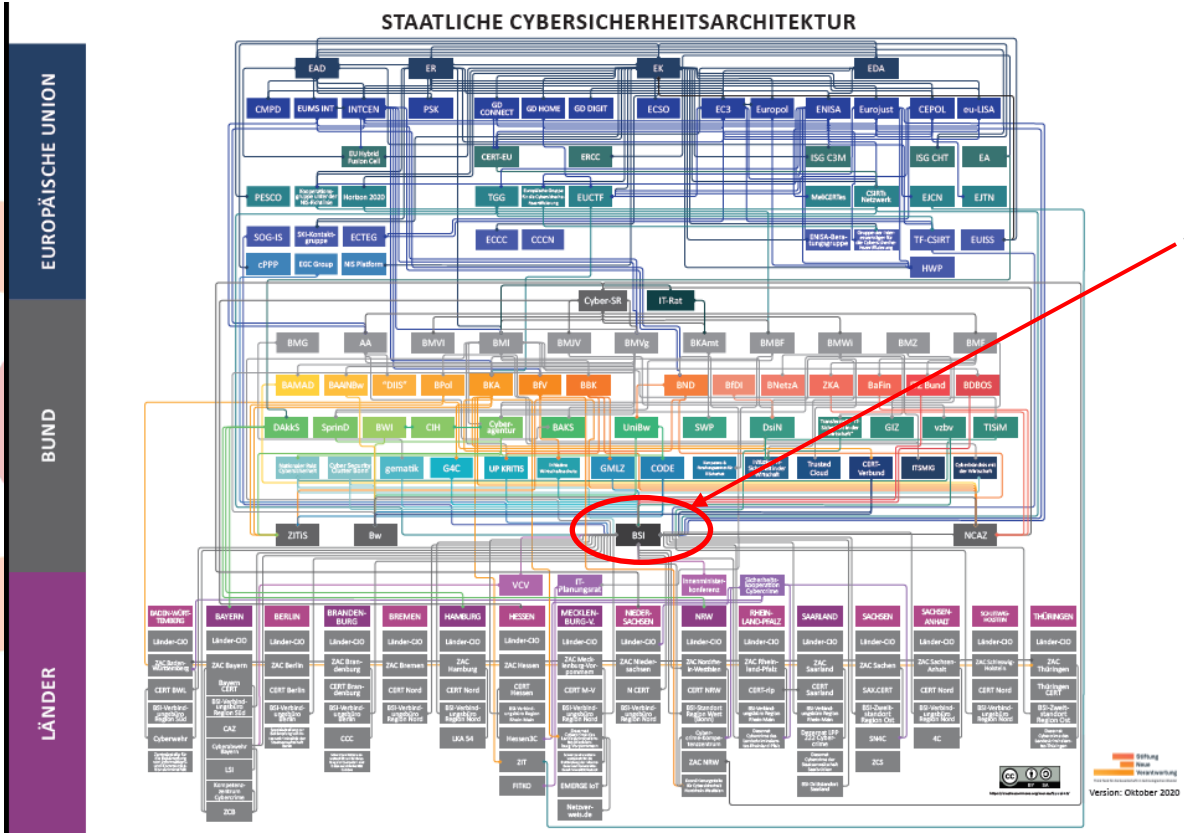
# 2 – Bedrohung: Gesetzliche Regelungen und Regulatorik

## Die gesetzliche „Bedrohungslage“ – Außen- und Innenwirkung

- Der Gesetzgeber unterscheidet grundsätzlich nicht, wie ein Unternehmen seine Informationsverarbeitung technisch und organisatorisch betreibt.
  - Die rechtliche Verantwortung ist nicht „outsource-bar“ oder „ver-cloud-bar“!
- Es gelten die üblichen Gesetze
  - EU Cybersecurity Act 2019/881 (die ENISA existiert seit 2004)
  - EU-DSGVO + (nachgeordnet) BDSG neu
  - EU-Richtlinie 2016/943 + Geheimnisschutz-Gesetz (GeschGehG)
  - BSI IT-Sicherheitsgesetz
  - EU ePrivacy-Richtlinie 2002 (Nachfolger in Vorbereitung, in D: „TTDSG“)
  - BGB und HGB und darauf basierende Regelwerke
    - z.B. GOBD, AO
  - Strafrecht, z.B.
    - § 202a StGB – Ausspähen von Daten
    - § 303b StGB – Computersabotage

# 2 – Bedrohung: Gesetzliche Regelungen und Regulatorik

## Staatliche Cybersicherheits-Organisationen in Europa und Deutschland




Quelle:
Stiftung Neue Verantwortung,
Oktober 2020







cat out

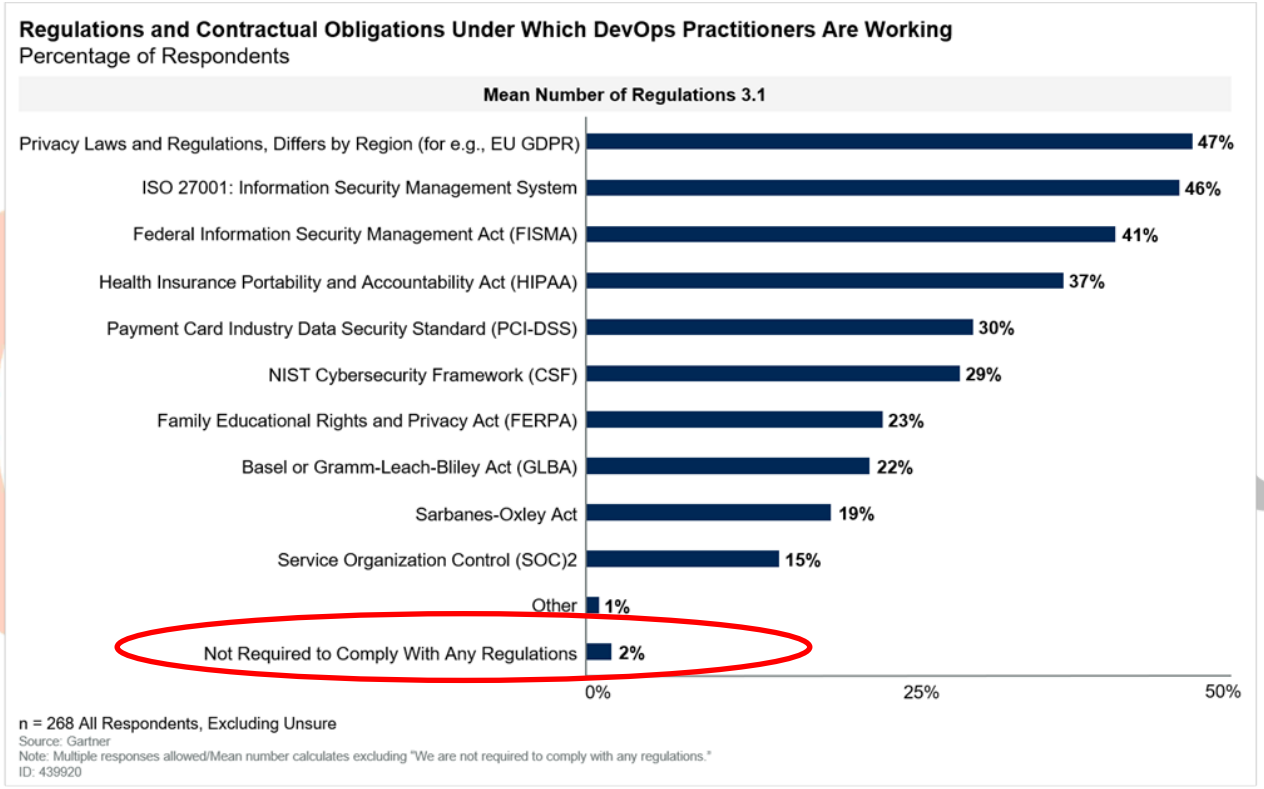
# 2 – Bedrohung: Regulatorik und Standards

## Weitergehende Vorgaben für Branchen und Produkte

- Für Finanzdienstleister:
  - BAIT / VAIT / KAIT
  - Umsetzung der MaRisk – Mindestanforderungen an das Risikomanagement
  - Basis: Kreditwesengesetz KWG, im Wesentlichen § 25 a, b
- Für viele Industrien spezifische Normen und Standards
  - z.B. VDE/IEC 62443-x für Produktionsbetriebe
- DIN/ISO 15408 - „Common Criteria“
  - Internationaler Produkt-Security-Standard

# 2 – Bedrohung: Regulatorik und Standards

## Auswirkungen auf DevOps-Beteiligte




Gartner: “3 Steps to Ensure Compliance and Audit Success With DevOps”, 11 October 2019 - ID G00439920

# 2 – Bedrohungen: Die Sicherheitslage wird ernster

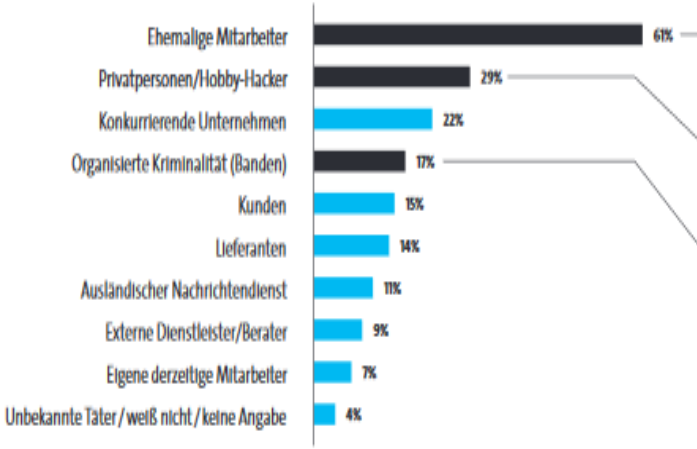
## Spektakuläre Angriffe nehmen zu und werden professioneller

- Neben den klassischen Angreifern („Hobby-Hacker“, Werkspionage, Geheimdienste) etablieren sich zunehmend regelrechte Cyber-Attack-Firmen
  - Dabei ist die Beeinträchtigung des Betriebs eines Unternehmens z.B. durch DDoS-Attacken o.ä. nicht mehr das eigentliche Ziel (vielleicht noch für Hobby-Hacker)
  - Das eigentliche Ziel sind Ihre Unternehmensdaten!
- Hauptintention der Angreifer:
  - (Kritische) Unternehmensdaten in ihre Hoheit bringen und die Originaldaten durch Verschlüsselung unbrauchbar machen („Ransomware“)
  - Dann Lösegeldforderungen stellen für:
    - Schlüssel zur Wiederherstellung ausliefern
    - Nichtveröffentlichung der gestohlenen Informationen, Vernichtungserklärung

# 2 – Bedrohungen: Die Sicherheitslage wird ernster

## Angreifer haben es ggf. schon bis in Ihr Unternehmen geschafft

- Externe Angreifer haben es ggf. schon geschafft, in Ihr Unternehmen einzudringen
  - Ausnutzen von Schwachstellen in Firewalls, Intrusion Detection,…
  - Phishing-Mails und Abgreifen von User Credentials
  - „Social Engineering“ zur Erlangung interner Informationen
- **Und manche waren / sind vielleicht sogar schon Ihre „Kolleginnen/en“!**
  - Einschleusen auf zentrale IT-Funktionen wie Administratoren, aber auch Entwickler etc. !
  - BKA: Monitoringbericht „Innentäter in Unternehmen 2“, Februar 2020



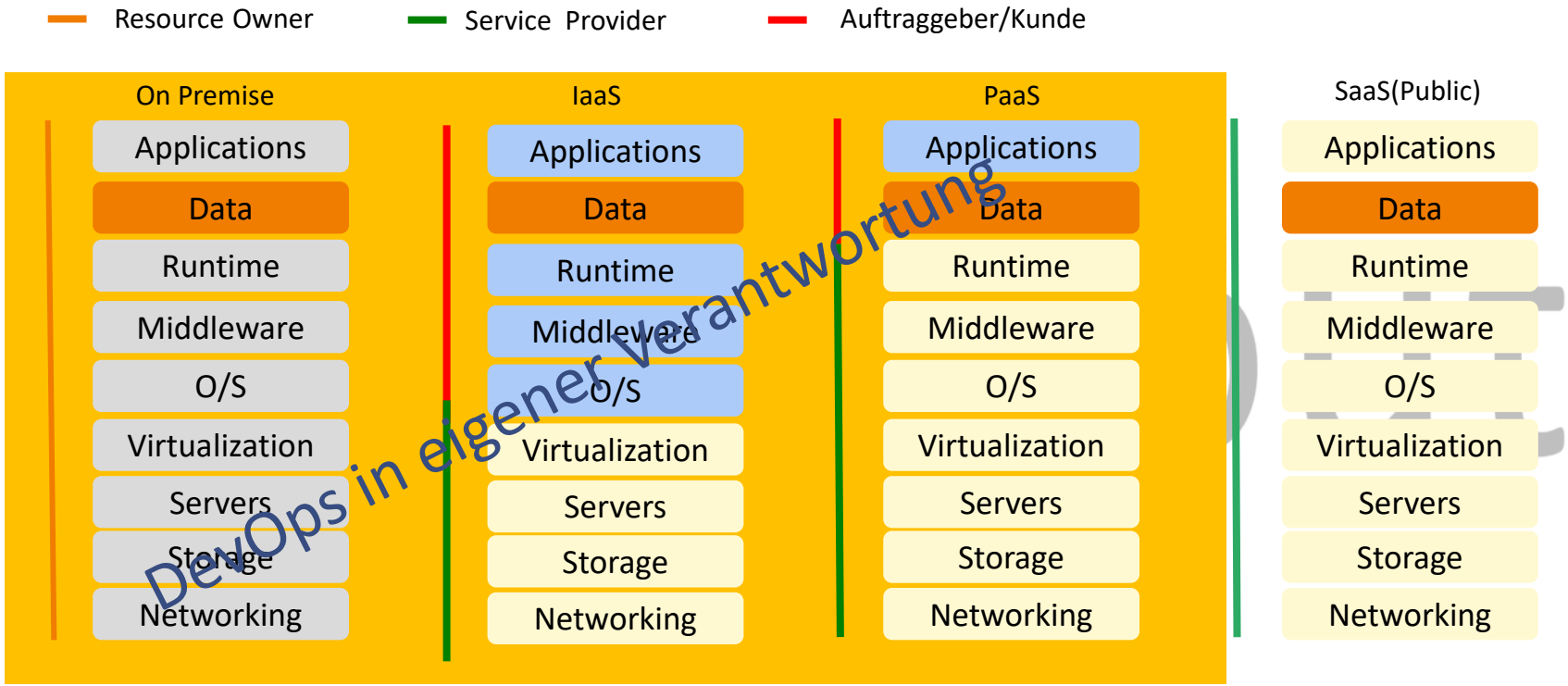
# 2 – Bedrohungen: Die Infrastruktur wird immer heterogener

## Aktuelle Bereitstellungsmodelle

- **On-Premise –** Infrastruktur wird in einem eigenen oder ausgelagerten Data Center genutzt.
- **Private Cloud / Outsourcing-** Infrastruktur wird zur ausschließlichen Nutzung einer einzigen Institution betrieben, kann von Ihr selbst oder einem Dritten geführt werden.
- **Public Cloud** – Infrastruktur wird für die offene Nutzung durch die Allgemeinheit oder z.B. einer ganzen Branche von einem Anbieter bereitgestellt.
- **Community Cloud** – eine mandanten-fähige Infrastruktur, die von einer Gruppe von Institutionen mit den gleichen Anforderungen gemeinsam genutzt wird.
- **Hybrid Cloud** – Infrastruktur besteht aus zwei oder mehr unterschiedlichen Cloud-Infrastrukturen (private, community oder public), verbunden durch Technologien, standardisierte Schnittstellen, die Daten- und Anwendungsportabilität ermöglichen.
- **Multi Cloud** – Nutzung mehrerer Cloud-Infrastrukturen gleichen Typs.

# 2 – Bedrohung: Verantwortungsverteilung

## Auftragnehmer - Auftraggeber




Kontrollmechanismen: Verantwortlichkeit im Innenverhältnis, Dritten gegenüber bleibt der Subscriber verantwortlich

# 2 – DevOps-Security

## DevOps als Angriffs-Einfallstor zur Produktion

- Die **Infrastruktur** zur Verbindung von Entwicklung und Produktion
  - Netzwerkabsicherung
  - Authentifizierung  und Autorisierung
- Der **Prozess** und die **Tool-Kette**
  - Einschleusen von Komponenten
  - Manipulieren von Tool-Parametern
- Die **Software-Komponenten** selbst
  - Eigene – unbemerkt manipulierte – Software
    - Bewusst Qualitätsprobleme nicht beseitigt/eingebaut
    - Backdoor-Code, der in der Produktion ausgenutzt werden kann
  - Open Source-Komponenten
    - Wer prüft explizit alle Qualitätsdimensionen?
    - Nutzung nimmt stark zu!

# 3 – „Zero Trust“

## Grundlegender Ansatz

- Das alte (Netzwerk-) Sicherheitsgrundkonzept

  “We want our network to be like an M&M, with a hard crunchy outside and a soft chewy center.”

  trägt nicht mehr!

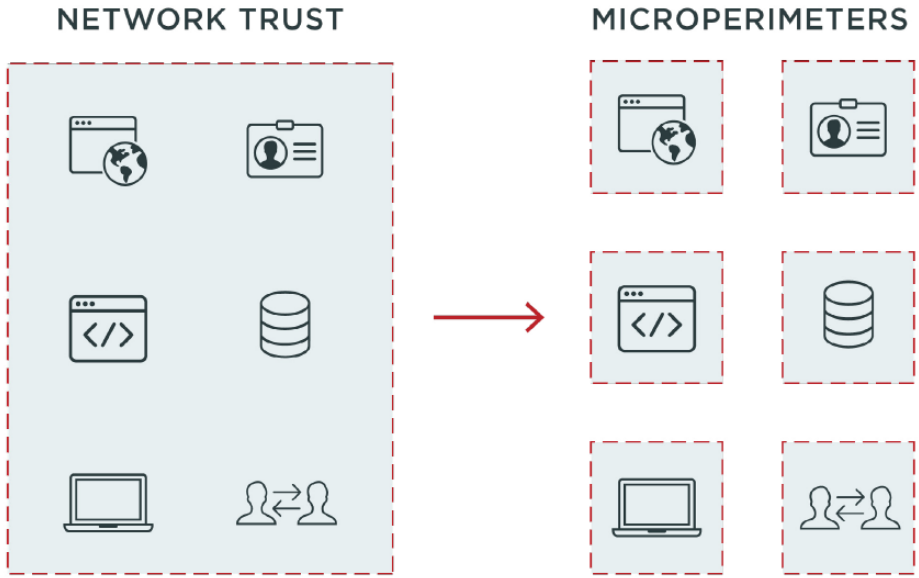

Quelle: mms.com

- Erste Ansätze zur Auflösung des alten Perimeter-Modells bereits in den 2000er Jahren durch DISA – „Black Core“
- Forrester (Kindervag et al.) ab 2011: Begriff „Zero Trust“
  - Einreichung als Vorschlag auf NIST-RfI „Cybersecurity Framework“ im Jahr 2013
- Seitdem hat Forrester das Konzept weiter ausgebaut
  - „Zero Trust eXtended Ecosystem Platform“

# 3 – Zero Trust

## Grundlegender Ansatz

- Der klassische Perimeter „Die Burg mit Wassergraben“ wird durch viele Perimeter ersetzt
- Zero Trust-Prinzipien (NIST 2013):
  - Ensure all resources are accessed securely regardless of location.
  - Adopt a least privilege strategy and strictly enforce access control.
  - Inspect and log all traffic.
- Niemand ist mehr per se für alle Zugriffe vertrauenswürdig, nur weil er sich einmal authentifiziert hat.
  - Rollenbasierte, wiederkehrende Prüfung bei jedem Zugriff




Quelle: pingdidentity.com

# 3 – UMSETZUNG VON ZERO TRUST

## KERNELEMENTE DER UMSETZUNG

- Für jedes Segment der IT-Infrastruktur natürlich weiterhin
  - Klassische Absicherung durch Netzsegmentierung, Firewalls, Intrusion Detection, Endpoint Detection, Virenabwehr, Phishing-Abwehr
- Zusätzlich für jeden Teil-Perimeter:
  - Authentifizierung und Autorisierung der Benutzer (auch „technischer“)
    - IGA –> IAM, PAM und Access Management
  - Authentifizierung aller Geräte und Applikationen (auch „externer“)
  - Bewertung jeder einzelnen Transaktion in Bezug auf ihre Vertrauenswürdigkeit („UEBA“ – User and Entity Behavioral Analysis)
  - Permanente Anpassung an aktuelle Bedrohungen („Threat Intelligence“)

# 3 – Zero Trust Authentifizierung/Autorisierung

## IGA – Identity Governance & Administration

- IAM (Identity and Access Management)
  - Identity- und Access-Management-Systeme als zentrale Authentifizierungsplattform
  - Identifikation der Benutzer und Kontrolle des Zugriffs auf zugeordnete Ressourcen
  - Single Sign-On (SSO)
  - Authentifizierung über Multifaktorauthentifizierung (MFA)
- **PAM** (Privileged Account Management)
  - Absicherung der besonders privilegierten Nutzer wie z.B. Administratoren
  - Aber auch Verwaltung der sog. „technischen Service Accounts“
    - Für API-Absicherung und **alle automatisierten Prozesse (wie auch DevOps)** relevant!
- Access Management
  - Verwaltung der Zugriffsrechte pro Ressource („RBAC“ – Roll-Based Access Control)

# 3 – DevSecOps
## Wesentliche Themengebiete

- Umsetzung von DevSecOps
  - Netzwerk- und Nutzermanagement für Entwicklungsumgebungen
  - Configuration Management und Software Composition Analysis unerlässlich
    - Eigene Komponenten können manipuliert sein
    - Aber vor allem „Open Source“-Einsatz
      - Untersuchung: Jede 8. Komponente enthält sicherheitstechnische Schwachstellen
  - Build-Prozesse
    - Statische Analyse und automatisierte Tests in Bezug auf Sicherheit der Software
  - Deployment
    - Absicherung durch Compliance
      - „Software Defined Compliance“ / „Compliance as Code“
      - Machine Learning-Ansätze zur Bewertung der Vertrauenswürdigkeit

# 3 – DevOps-Security

## DevSecOps als wesentliche Ergänzung

| | Integration Points and Degree of Automation | | | | |
|---|---|---|---|---|---|
| DevSecOpsTooling | Design | Development (IDE) | Repository Manager | CI/CD | Post-Deployment |
| Open source governance | ● | ● | ● | ● | ● |
| Open source software analysis | ● | ● | ● | ● | n/a |
| Static Application Security Testing (SAST) | ● | ● | ● | ● | n/a |
| Dynamic Application Security Testing (DAST) | ● | n/a | n/a | n/a | ◔ |
| Interactive Application Security Testing (IAST) | ● | n/a | n/a | ● | n/a |
| Mobile Application Security Testing (MAST) | ◐ | n/a | ◐ | ◐ | n/a |
| Run-time Application Self Protection (RASP) | n/a | n/a | n/a | ◐ | ● |
| Container and Infrastructure Security | ◐ | n/a | ● | ● | ● |

Quelle: Gartner, December 2017 - "Structuring Application Security Practices and Tools to Support DevOps and DevSecOps"

# 3 – DevOps-Security

## Beispiel-Metriken der GSA (U.S. General Services Administration)

- High Value Metrics



| Metric | Description | Associated Domain(s) |
|---|---|---|
| Deployment frequency | Number of deployments to production in a given time frame | Application Deployment; Authority to Operate Processes |
| Change lead time (for applications) | Time between a code commit and production deployment of that code | Overarching; Authority to Operate Processes; Patch Management |
| Change volume (for applications) | Number of user stories deployed in a given time frame | Overarching |
| Change failure rate | Percentage of production deployments that failed | Application Deployment |
| Mean time to recovery (MTTR) (for applications) | Time between a failed production deployment to full restoration of production operations | Application Deployment; Backup and Data Lifecycle Management; Patch Management |
| Availability | Amount of uptime/downtime in a given time period, in accordance with the SLA | Availability and Performance Management; Network Management |
| Customer issue volume | Number of issues reported by customers in a given time period | Overarching |
| Customer issue resolution time | Mean time to resolve a customer-reported issue | Overarching |
| Time to value | Time between a feature request (user story creation) and realization of business value from that feature | Overarching; Authority to Operate Processes |
| Time to ATO | Time between the beginning of Sprint 0 to achieving an ATO | Overarching; Authority to Operate Processes |
| Time to patch vulnerabilities | Time between identification of a vulnerability in the platform or application and successful production deployment of a patch | Authority to Operate Processes |

Quelle: GSA - https://tech.gsa.gov/guides/dev_sec_ops_guide/

# 3 – DevOps-Security

## Beispiel-Metriken der GSA (U.S. General Services Administration)

- Supporting Metrics (Ausschnitt)

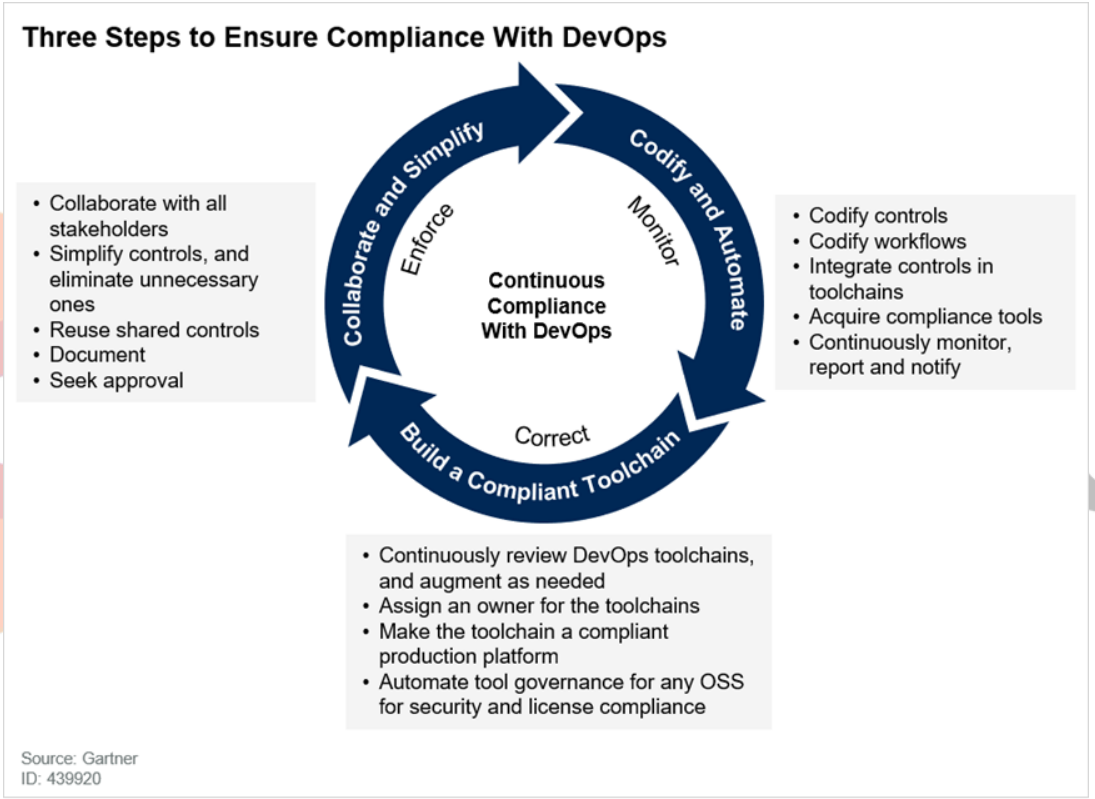


| Metric | Description | Associated Domain(s) |
|---|---|---|
| Test coverage | Percentage of code that is covered by automated tests | Application Development, Testing, and Operations |
| Change types | Percentage of features vs fixes vs security patches | Change Management; Patch Management |
| Time to availability of event information | Time from an event to information about the event being available to the DevSecOps team or end users | Logging, Monitoring, and Alerting |
| Developer onboarding | Time from a developer joining the team to ability to commit code for production deployment | Application Development, Testing, and Operations |
| Change resolution time | Time between a change proposal and closing (implementation or rejection) | Platform Governance |
| Change lead time (for the DevSecOps platform) | Time between a change (e.g., code commit) and platform deployment of that change | Change Management; Patch Management; Network Management; Platform Governance |
| Change volume (for the DevSecOps platform) | Number of user stories deployed in a given time frame | Change Management; Platform Governance |
| Change failure rate (for the DevSecOps platform) | Percentage of platform deployments that failed | Change Management; Platform Governance |
| Mean time to recovery (MTTR) (for the DevSecOps platform) | Time from a failed platform deployment to full restoration of platform operations | Change Management; Patch Management; Platform Governance |
| Change lead time for images | Time from identification of need for a new/updated image to its availability for production use | Image Management |
| Image publishing frequency | Number of new/updated images published in a given time frame | Image Management |
| Logging availability | Amount of uptime/downtime of the logging system in a given time period | Logging, Monitoring, and Alerting |

Quelle: GSA - https://tech.gsa.gov/guides/dev_sec_ops_guide/

# 3 – DevOps-Compliance

## Permanente Überprüfung aller DevOps-Security-Massnahmen




Gartner: “3 Steps to Ensure Compliance and Audit Success With DevOps”, 11 October 2019 - ID G00439920

# 4 – Fazit und Ausblick

## DevOps und Zero Trust - Sicherheit und Compliance

- DevOps wird sich
  - gerade im Zuge der Cloud- und Microservices-Ausweitung

  weiter verbreiten
- Die regulatorischen Vorbehalte lassen sich mitigieren
  - Weitestgehend automatisierte Compliance-Absicherung
  - Last line of defense: menschliche Kontrolle zumindest der Zweifelsfälle
- Die „Unbekümmertheit“ in Bezug auf die Sicherheit von/in Entwicklungssystemen muss noch abnehmen
  - Gerade in Bezug auf Datenschutz – viele Testdaten immer noch „Live-Daten“
- Zero Trust bleibt ein wesentliches Architektur- und Kulturmodell
  - auch im Bereich Software-Entwicklung und -Deployment

# FRAGEN, KOMMENTARE ??

Quelle: https://pixabay.com/de/vectors/mikrofon-audio-musik-h%C3%B6ren-159768/

# KONTAKT