KONTINUIERLICHE SICHERHEIT IN DER CLOUD

19. BSI Cyber-Sicherheits-Tag

7.11.2017 - Stuttgart

Fraunhofer AISEC

Clouditor
Continuous Cloud Assurance

EUSEC
EU SECURITY CERTIFICATION

Fraunhofer AISEC

- Gegründet 2009 als Projektgruppe des Fraunhofer SIT
- Eigenständige Einrichtung seit 2011, Institut seit 2013
- Anzahl der Mitarbeiter ca. 100
- Enge Anbindung an TU München
  - Fakultät für Informatik (Prof. Eckert)
  - Fakultät für Elektrotechnik und Informationstechnik (Prof. Sigl)
- Finanzierung (Fraunhofer-Modell)
  - Bis zu 20% staatlich
  - 80% durch öffentliche Forschungsprojekte und Auftragsforschung (Industrie)

Cloud-Sicherheit am Fraunhofer AISEC

- Forschungsprojekte
  - Anfänge: BSI-Studie zu Cloud-Sicherheit (2009)
  - *NGCert* (BMBF, 2014-2017, www.ngcert.de) – kontinuierliche Cloud-Zertifizierung
  - *EU-SEC* (EU H2020, 2017-2020, www.sec-cert.eu) – Europäisches Rahmenwerk für Cloud-Zertifizierungen
  - *Bayern-Cloud* (StWMi Bayern, 2017-2020) - Sichere Community-Clouds für regionale Anbieter
  - Zahlreiche Industrieaufträge (Sicherheitsanalysen, Konzepte, Workshops, ...)
- Labor-Umgebungen
  - Private Cloud OpenStack-Umgebung mit Kubernetes-Cluster
  - Testinfrastrukturen in AWS, Azure
- Tools: *Clouditor* (www.clouditor.de)

Warum ist kontinuierliche Sicherheit notwendig?

- Veränderung!
  - **... der Infrastruktur**
  - ... der Konfigurationen
  - ... des Stands der Technik
  - ... der angebotenen Dienste
  - ... der geographischen Lokation
  - ... der zugrunde liegenden Dienste
  - ...

Beispiel: Automatisch skalierbare Systeme

Moderne Cluster-Systeme passen sich teilweise an Auslastung und Benutzung an, dadurch verändert sich automatisch die zugrunde liegende Infrastruktur

Warum ist kontinuierliche Sicherheit notwendig?

- Veränderung!
  - ... der Infrastruktur
  - **... der Konfigurationen**
  - ... des Stands der Technik
  - ... der angebotenen Dienste
  - ... der geographischen Lokation
  - ... der zugrunde liegenden Dienste
  - ...

Beispiel: Komplexität und Abhängigkeiten

Geringe Konfigurationsänderungen können oft verheerende Folgen haben wenn die Auswirkung auf andere Systeme unterschätzt wird

Warum ist kontinuierliche Sicherheit notwendig?

- Veränderung!
  - ... der Infrastruktur
  - ... der Konfigurationen
  - **... des Standes der Technik**
  - ... der angebotenen Dienste
  - ... der geographischen Lokation
  - ... der zugrunde liegenden Dienste
  - ...

Beispiel: Verbindungsverschlüsselung

Gerade die Sicherheit von TLS-Verbindungen unterliegt einem steten Wandel (z.B. benutzte Cipher-Suites)

Warum ist kontinuierliche Sicherheit notwendig?

- Veränderung!
  - ... der Infrastruktur
  - ... der Konfigurationen
  - ... des Stands der Technik
  - **... der genutzten Dienste**
  - ... der geographischen Lokation
  - ... der zugrunde liegenden Dienste
  - ...

Beispiel: Nutzung eines einen neuen Cloud-Dienstes

Technisch „sofort“ möglich, es muss aber gewährleistet sein, dass die bisherigen Sicherheitsrichtlinien auch für den neuen Dienst greifen

Warum ist kontinuierliche Sicherheit notwendig?

- Veränderung!
  - ... der Infrastruktur
  - ... der Konfigurationen
  - ... des Stands der Technik
  - ... der angebotenen Dienste
  - **... der geographischen Lokation**
  - ... der zugrunde liegenden Dienste
  - ...

Beispiel: Geo-Replikation

Bewusste Replikation von Daten oder virtuellen Maschinen in verschiedene geographische Regionen kann die Ausfallsicherheit erhöhen. Eine fehlerhafte Konfigurationen kann allerdings Problem verursachen, wenn Daten Rechtsräume verlassen (z.B. die EU)

Warum ist kontinuierliche Sicherheit notwendig?

- Veränderung!
  - ... der Infrastruktur
  - ... der Konfigurationen
  - ... des Stands der Technik
  - ... der angebotenen Dienste
  - ... der geographischen Lokation
  - **... der zugrunde liegenden Dienste**
  - ...

Beispiel: Nutzung von SaaS/PaaS

Platform-as-a-Service bringt eine kostensparende Abstraktion, allerdings entziehen sich Veränderungen des Dienstanbieters oft der Kontrolle des Nutzers

Was ist kontinuierliche Sicherheit?

- Das **Aufrechterhalten** eines **Sicherheitsniveaus** über die **gesamte Lebenszeit** eines Dienstes
- Technisch: Kontinuierlicher Abgleich des Ist-Zustands mit einem Soll-Zustand
- Drei große Herausforderungen
  - Was ist „kontinuierlich"?
  - Was ist eigentlich der gewünschte Soll-Zustand?
  - Wie und wo überprüfe ich den Ist-Zustand?

Was ist „kontinuierlich"?

- Aus der Wissenschaft
  - Abgrenzung diskret (Anzahl von Personen) vs. kontinuierlich/stetig (Regenbogenfarben)
- Aus der Praxis
  - Annäherung an kontinuierlich, jedoch mit realistischem Grundverständnis
  - Typischerweise Intervalle von Überprüfungen im Bereich von Minuten oder Stunden
  - Empfohlene Intervalle richtet sich stark nach der abzuprüfenden Sicherheitseigenschaft

Was ist „kontinuierlich"? – Beispiel

- Kriterium
  - Ein Cloud-Provider empfiehlt Kunden eine Rotation der API-Zugangsschlüssel alle 90 Tage
- Tageweises Überprüfen der Rotation ausreichend
- Kürzeres Abprüfen bietet keinen offensichtlichen Mehrwert

Was ist der Soll-Zustand?

- Liste von **automatisierbar** prüfbaren **technischen Anforderungen**
- Technische Anforderungen adressieren
  - High-Level Sicherheitsanforderungen, z.B. OWASP Cloud Top 10 Security Risks
  - Gesetze, Rahmenwerke, z.B. EU-DSGVO
  - Zertifikate/Kriterienkataloge, z.B. C5, CSA CCM
  - Unternehmensweite Sicherheitsrichtlinien
  - Best Practices, z.B. Center for Internet Security (CIS) Benchmarks
- Herunterbrechen von Kriterienkatalogen auf technische Anforderungen ist nicht trivial
  - Problem des “Semantic Gap”
  - Technische Konkretisierung ist notwendig

Beispiel “Verschlüsselung” – Anforderungen aus Kriterienkatalogen

- EU-DSGVO fordert in Art. 32. (1) a) „Verschlüsselung personenbezogener Daten“
- C5 KRY-02 - Verschlüsselung von Daten bei der Übertragung über öffentliche Netze
  - BSI Stand der Technik: TLS 1.2 mit Perfect Forward Secrecy
  - Verweis auf TR-02102-2; expliziter Ausschluss von SSL (3.0)
- C5 KRY-03 - Verschlüsselung von sensiblen Daten bei der Speicherung
  - In KRY-01 wird die Verwendung von AES empfohlen
- CSA CCM EKM-03 - Encryption & Key Management Sensitive Data Protection
- Abstraktion: “Daten müssen auf dem Weg in die Cloud nach dem Stand der Technik verschlüsselt werden. Sensible Daten müssen auch in der Cloud selbst verschlüsselt gespeichert werden. Falls vorhanden, sollte ein Key Management System des Cloud-Providers benutzt werden”

Beispiel “Verschlüsselung” – Detaillierung

- Was sind sensitive Daten?
  - Kundendaten, Audit-Logs, Finanztransaktionen, ...
- Identifikation von Datenspeichern mit sensitiven Daten
  - S3 Buckets, EBS Volumes, RDS Datenbanken, (Azure Storage Accounts, Azure SQL), ...
- Identifikation von Einsprungs-Punkten
  - API Gateway, VMs mit Webserver, ...
- Gibt es einen Key Management Dienst?
  - Ja, Amazon KMS (Azure Key Vault)

Beispiel “Verschlüsselung” – Technische Abbildung (AWS)

- Empfohlen: Ressource Groups zur Kapselung der relevanten Objekte
- Alle S3 Buckets dürfen keine unverschlüsselten Objekte enthalten
- Alle verschlüsselten S3 Objekte dürfen nur mit einem von gemanagten Schlüssel verschlüsselt worden sein
- Alle S3 Buckets enthalten eine Policy, welche das Hochladen von unverschlüsselten Objekten verhindert
- Alle EBS Non-Root Volumes sind verschlüsselt
- Alle RDS Datenbanken sind verschlüsselt
- Alle TLS Endpunkte benutzen TLS 1.2 und Cipher Suites mit Perfect Forward Secrey
- Alle Zertifikate haben eine valide Vertrauenskette und geeignete Signatur-Algorithmen (z.B. SHA-256)
- ...

Wie und wo überprüfe ich den Ist-Zustand?

- Automatisierte Tools, z.B.
  - CIS Benchmark Checker von AWS Labs https://github.com/awslabs/aws-security-benchmark
  - Clouditor
  - Eigenentwicklungen
  - ...
- Überprüfungen sollten so wenig invasiv wie möglich sein
- Einige Eigenschaften (z.B. TLS-Sicherheit, OAuth-Sicherheit) lassen sich von außerhalb ohne zusätzlichen API-Zugriff testen
- Ein Großteil benötigt allerdings direkten Zugriff auf Cloud-Ressourcen, z.B. mittels API

Wie und wo überprüfe ich den Ist-Zustand?

- Deployment der Tools in der Regel in der gleichen Cloud-Infrastruktur
- Vorteile:
  - Credentials für API-Zugänge verlassen die Cloud nicht
  - AWS bietet beispielsweise „roles“ für VMs um diese automatisch mit API-Zugängen zu versehen
- Nachteile:
  - Tooling wird Teil des Dienstes (aus Sicherheits- und Risikosicht), muss ausreichend abgesichert sein

Clouditor

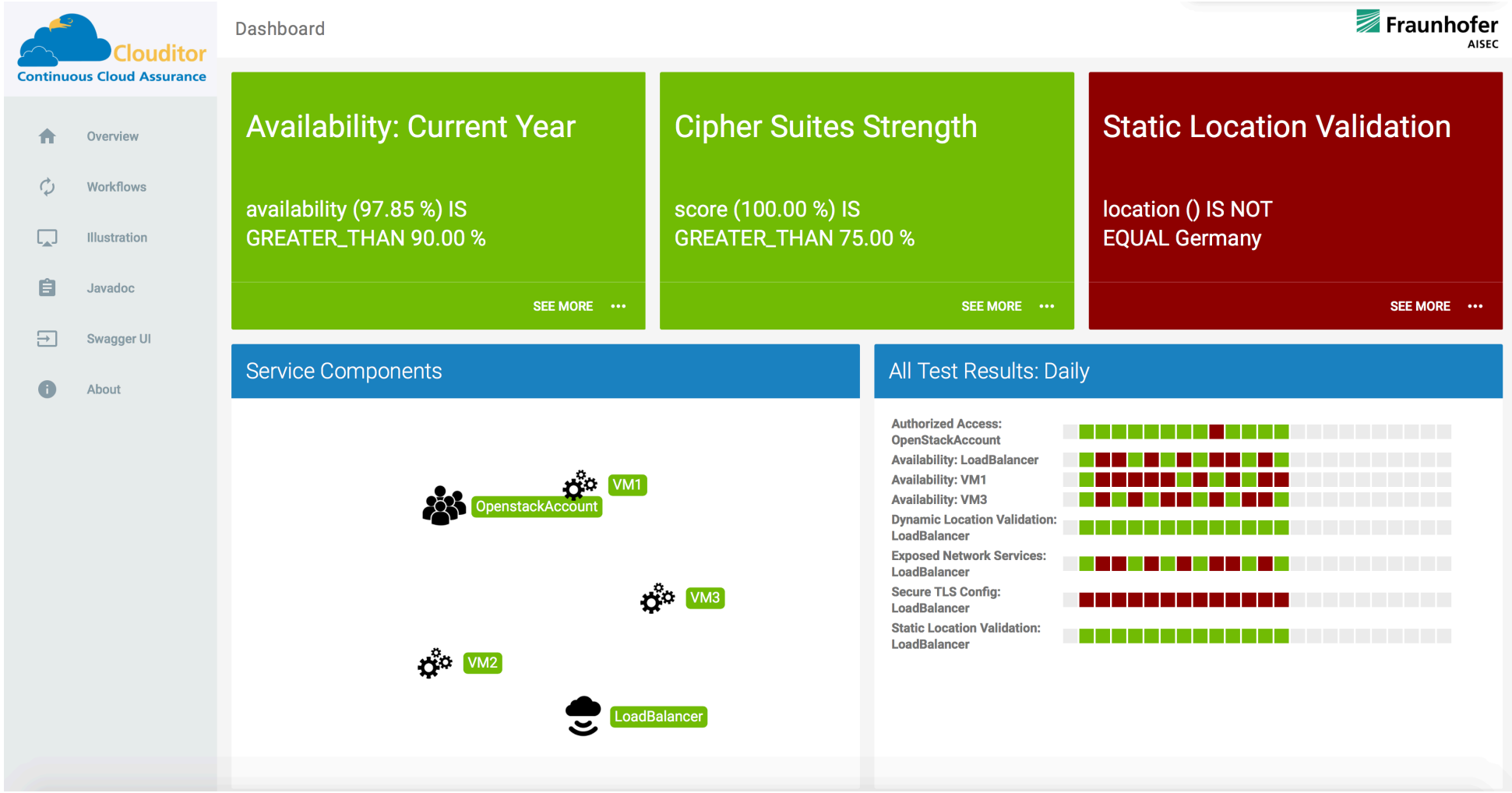
Continuous Cloud Assurance

Fraunhofer AISEC

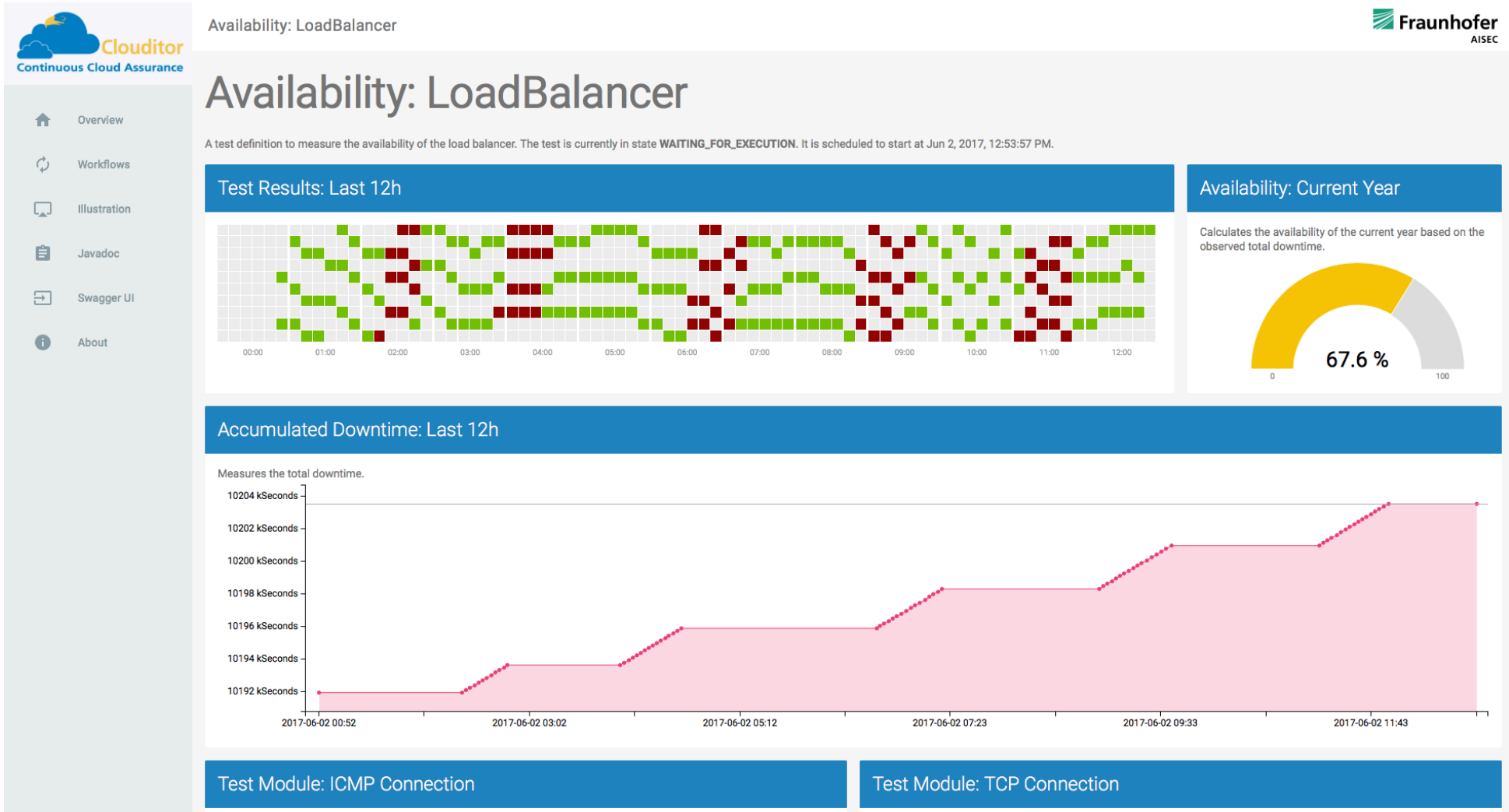
Clouditor – Continuous Cloud Assurance

- Aktuelles Entwicklungsprojekt von Fraunhofer AISEC: www.clouditor.de
- Tool zur automatisierten Abprüfung von Sicherheitseigenschaften (in der Cloud)
- Ca. 20 Test Module implementiert und weitere 20 in der Entwicklung
- Beispiele:
  - Überprüfung auf Verschlüsselung in AWS und Microsoft Azure
  - Überprüfung auf Nutzung von „Bring Your Own Key“ in Microsoft Azure
  - Sicherheit von TLS-Endpunkten
  - Validierung von Firewallregeln und Security Groups
  - Veränderung der Geo-Lokation von virtuellen Ressourcen
- Automatisiertes Service Discovery zur Unterstützung der Asset-Identifikation

Clouditor - Dashboard



Clouditor - Detailansicht



Zusammenfassung

- Cloud bedeutet konstante Veränderung und bedingt daher kontinuierliche Sicherheit
- Das Ziel ist das Aufrechterhalten eines Sicherheitsniveaus über die gesamte Lebenszeit eines Dienstes
- Zentraler Punkt ist das Aufstellen von technisch und automatisierbar prüfbaren Anforderungen
- Ableitung von Sicherheitsanforderungen aus Kriterienkatalogen oder Gesetzen ist nicht trivial
  - Nur ein Teil ist überhaupt automatisierbar - großer „Semantic Gap“ in vielen Kriterien
  - Stückweises Herunterbrechen und Konkretisierung auf technische Anforderungen notwendig
- Nachweis der technischen Anforderungen kann u.U. Unterstützung für ein Audit leisten (Forschungsgegenstand)
- Tooling kann den Prozess und die Durchführung unterstützen