codecentric

WHITEPAPER

Quo vadis, APM?

Einsatzgebiete, aktueller Stand und Blick in die Zukunft von Application Performance Management

Inhalt

Systemlandschaften von Unternehmen entwickeln sich in rasantem Tempo. Weil sich Applikationen, Services und deren Abhängigkeiten kaum noch manuell erfassen oder warten lassen, setzen viele Unternehmen auf eine systematische Überwachung per Application Performance Management (APM). Doch welche Lösungen gibt es hierfür? Was kann die neueste Generation von APM-Tools leisten? Warum scheitert die Implementierung von APM-Lösungen auf lange Sicht so häufig? Und woran erkennt man ein gutes APM-Werkzeug?

Dieses Whitepaper richtet sich an IT-Entscheider, DevOps Engineers, QA-Agents und alle Softwareentwickler, die sich mit der systematischen Überwachung verteilter Systeme befassen und die Performance-Überwachung ihrer Systeme automatisieren wollen. Wir beleuchten die Beziehung zwischen Softwarearchitekturen und Performance Management über die Jahre hinweg, adressieren Schwierigkeiten und klassische Probleme bei der Implementierung von APM-Lösungen und gewähren Einblicke in die neuesten Entwicklungen. Im Anhang fasst eine 11-Punkte-Checkliste die wichtigsten Anforderungen an eine kommerzielle APM-Lösung zusammen.

Executive Summary

Systemarchitekturen haben sich in den letzten Jahren radikal gewandelt, die Anforderungen an die IT ebenso. Mit stetig wachsenden Applikationslandschaften nehmen auch die Anfälligkeiten von Anwendungen und ihre Beziehungen untereinander stetig zu, das gesamte Konstrukt erfährt immer komplexere Auswüchse.

Microservices lösen zwar einerseits viele traditionelle Probleme in der Software Delivery, führen andererseits aber auch zu einem neuen Level an Abhängigkeiten.

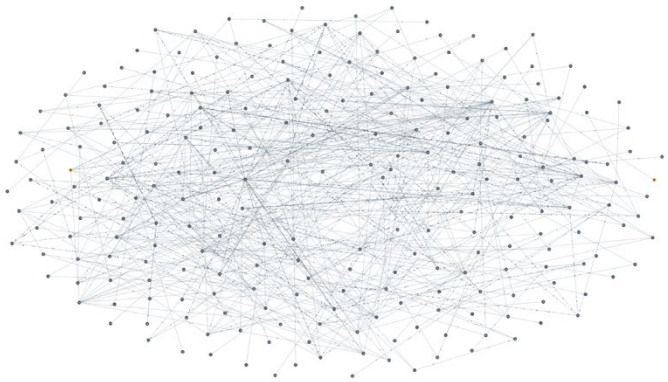
Durch moderne Application-Performance-Management-(APM)-Werkzeuge können IT-Landschaften systematisch überwacht und Fehler frühzeitig erkannt und behoben werden. Dabei leistet eine neue, KI-gestützte Generation von APM-Tools mithilfe von *Observability* große Hilfestellung beim effizienten Stack-Management.

Der Bedarf an intelligenten, mitlernenden Lösungen ist groß, doch die Tools unterscheiden sich zum Teil gravierend in Sachen Leistungsumfang, Integrierbarkeit und Kosten. Hier lohnt es sich, vorab einige Anforderungen an die Lösung zu formulieren, wie die 11-Punkte-Checkliste im Anhang es exemplarisch vorschlägt. Denn die Praxis zeigt: Viel zu oft werden teure Werkzeuge aufwendig implementiert und erweisen sich bald als Kostengrab, das aufgrund unvollständiger Monitoring-Daten oder schlechter Usability nicht gewinnbringend eingesetzt werden kann. Hier lohnt es sich, genau hinzuschauen und auch spezifische Beratungsangebote wahrzunehmen.

Ein Wettlauf gegen die eigenen Systeme

Die IT-Landschaften in deutschen Unternehmen sind im stetigen Wandel. Monolithische Systeme werden zunehmend durch verteilte Architekturen mit zahlreichen Microservices ersetzt, die mehr Flexibilität, Skalierbarkeit und Sicherheit durch Redundanzen ermöglichen. Daten und Nutzer*innen können sich prinzipiell überall be-

finden, während die Entwicklungsumgebungen durch neue Trends wie SaaS, hybride Netzwerke, Cloud Native und zunehmende Mobilität immer anspruchsvoller werden.



Mit steigender Komplexität nehmen jedoch auch die Abhängigkeiten zu; das große Ganze wird dabei immer schwieriger zu durchdringen. Kommt es zu Ausfällen einzelner Teilsysteme, müssen nicht selten alle verknüpften Services einzeln unter die Lupe genommen werden, um das Problem zu finden, zu verstehen und zu lösen. Im schlimmsten Fall kämpfen sich dabei erfahrene, hochqualifizierte DevOps Engineers über Tage und Wochen durch sämtliche Systemkomponenten, um ein einziges Problem aufzuspüren, das die Performanz des gesamten Systems lähmt.

Vom Cost Center zum USP

Auch die Wahrnehmung von IT im Unternehmenskontext hat sich in den letzten Jahren stark gewandelt. Wurde sie in den Chefetagen vor einiger Zeit lediglich als Mittel zum Zweck — teilweise sogar als reines Cost Center — wahrgenommen, hat mittlerweile ein Paradigmenwechsel stattgefunden. Viele Geschäftsführer verste-

hen unlängst die geschäftskritische Relevanz einer robusten IT-Infrastruktur; immer mehr Geschäftsmodelle basieren zudem heute auf digitalen USPs. Kommt es in der Performance der eingesetzten Systeme zu Engpässen, wird der unmittelbare Zusammenhang zwischen Systemstabilität und Wettbewerbsvorteil auf schmerzliche Weise deutlich.

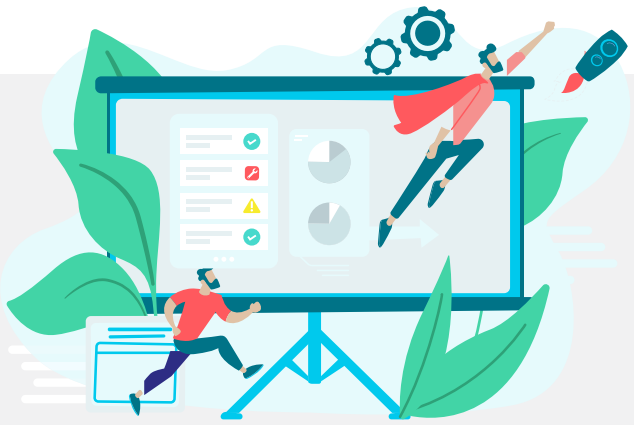
Hier entsteht in vielen IT-Abteilungen der Bedarf nach intelligenten Werkzeugen, die die firmeneigene IT-Landschaft systematisch überwachen und Probleme automatisch und frühzeitig erkennen. An diesem Punkt setzen APM-Lösungen an.

APM ist der Schlüssel, um Probleme mit der Anwendungs-Performance auf allen Ebenen frühzeitig zu verstehen und hilft, die Applikationslandschaft systematisch und automatisiert zu überwachen. Nur, wenn die IT-Abteilung alle eingesetzten Apps kennt, kann sie Flaschenhälse auch innerhalb des Anwendungscodes und der Anwendungssysteme erkennen, geschlossene Service Levels einhalten und eine solide Geschäftsgrundlage sicherstellen.

System: 1 – APM: 0

Vergleicht man jedoch die Entwicklung der Anwendungsüberwachung, so wird schnell deutlich, dass diese der Entwicklung der eingesetzten Technologien und Systemlandschaften in den letzten zwei Dekaden regelmäßig hinterhergehinkt ist. Kaum war es einer neuen Generation von APM-Lösungen möglich, mit dem nächsten Entwicklungsschub der Anwendungslandschaft gleichzuziehen, hängten Weiterentwicklungen im Technologie-Stack die Systemüberwachung schon wieder ab.

Was also muss ein modernes Application-Performance-Management-System mittlerweile leisten können? Wie wird es richtig eingesetzt und wie kann vermieden werden, dass es schneller auf dem Abstellgleis steht, als es implementiert wurde?



Exkurs: Der Wandel der IT-Landschaften in den letzten 20 Jahren

Betrachtet man die Evolution von Monitoring-Systemen, muss zunächst verstanden werden, dass jeder Ansatz eine Antwort auf die jeweils vorherrschenden technologischen Gegebenheiten war. Wie Geschäftsmodelle und Entwicklungsumgebungen auch, sind APM-Werkzeuge mit ihren Herausforderungen gewachsen. Es lohnt sich daher, zunächst einen Blick auf den infrastrukturellen Kontext zu werfen, um Sinn und Zweck der entwickelten und eingesetzten Performance-Anwendungen zu verstehen. Hierbei lassen sich drei Zeitalter identifizieren:

1. Monolithische Architekturen

Zur Jahrtausendwende waren große, in sich geschlossene Softwaresysteme der aktuelle State of the Art. Diese Systeme, wie z. B. Mainframes und Client-Server-Architekturen, waren in der Regel nicht in Teilsysteme gegliedert oder auf Basis einzelner, zusammen wirkender Komponenten erstellt. Sie waren weder wartbar noch erweiterbar, da die einzelnen Komponenten dieser Systeme nur mit erheblichem Aufwand modifiziert und an neue Bedingungen angepasst werden konnten.

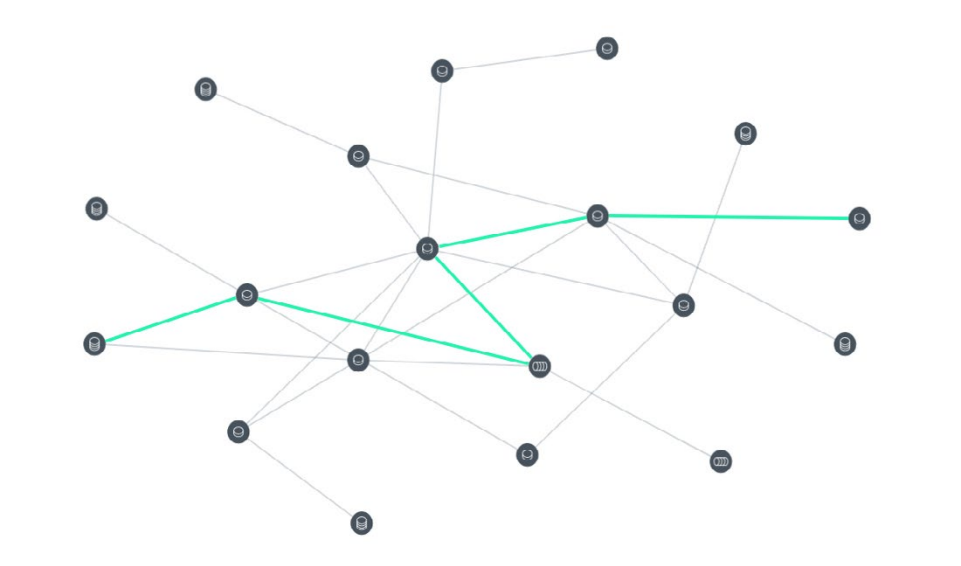
Um Fehler und Leistungsprobleme aufzuspüren, wurden *Thin-Client-Architekturen* geschaffen und einzelne Teilbereiche — sogenannte Silos — der IT-Umgebung überwacht. So konnten erstmals mithilfe kommerzieller Lösungen oberflächliche Einsichten in einzelne Applikationen stattfinden (Monitoring). Ursachenanalysen zu auftretenden Problemen wurden mit für damalige Verhältnisse hoch entwickelter Transaktionsüberwachung über mehrere JVMs (Java Virtual Machines) ausgeführt. Auf diese Weise war es Entwicklern und Systemadministratoren zwar prinzipiell möglich, ihre Systeme systematisch zu überwachen, jedoch waren die Einblicke selektiv, statisch und nicht automatisiert.

2. Verteilte Applikationen

Mit Beginn des nächsten Jahrzehnts änderte sich die Sichtweise auf die Architektur von Softwaresystemen. Software wurde nicht mehr als zentraler ‚Big Bucket' verstanden, sondern zunehmend in kleinere Anwendungen zerteilt: Service-orien-

ted Architectures (SOA) versprachen ein Wachstum an Flexibilität und Agilität. Die entstandenen Client-Server-Architekturen änderten sich mit immer größerer Geschwindigkeit, was die Ursachenforschung bei Performance-Problemen und zahlreichen anderen Fragestellungen äußerst mühsam machte.

Der damalige Ansatz: Monitoring einzelner Business-Transaktionen und Rich-Client- (häufig auch ‚Fat-Client'-)Architekturen. *Distributed Tracing* gewann an Bedeutung und *Baselining* wurde zur Erkennung von Antwortzeitverhalten eingesetzt. Durch eine höhere Granularität der Ergebnisse und frühzeitige(re)s Erkennen von Fehlern konnten die MTTR (Mean Time To Repair) deutlich verbessert und Probleme so kostengünstiger behoben werden. Auch ein verbessertes End-to-End-Monitoring, also das Verfolgen der Aufrufe vom Client bis zum Server, war hierdurch möglich.
Trotz des Quantensprungs standen die Betreiber solcher Infrastrukturen weiterhin vor beträchtlichen Problemen:



- Die Ergebnisse des Monitorings beruhten auf Stichproben (Sampling) und lieferten daher nicht nur ungenaue, sondern teilweise auch veraltete Daten zu den Fehlerursachen.
- Der Aufwand für die immer komplexer werdenden Frameworks wuchs deutlich schneller als das Vermögen einer ausreichenden Instrumentierung. Die Anforderungen rannten den Lösungsansätzen also förmlich davon.
- Der Bedarf an Instrumentierung wurde zum Overhead. Bei erkannten Risiken schaltete sich die Überwachung daher eigenständig aus.
- Konfiguration und Betrieb der Werkzeuge erforderten Experten-Know-how und konnte nur durch eingehend geschulte Anwender vorgenommen werden.

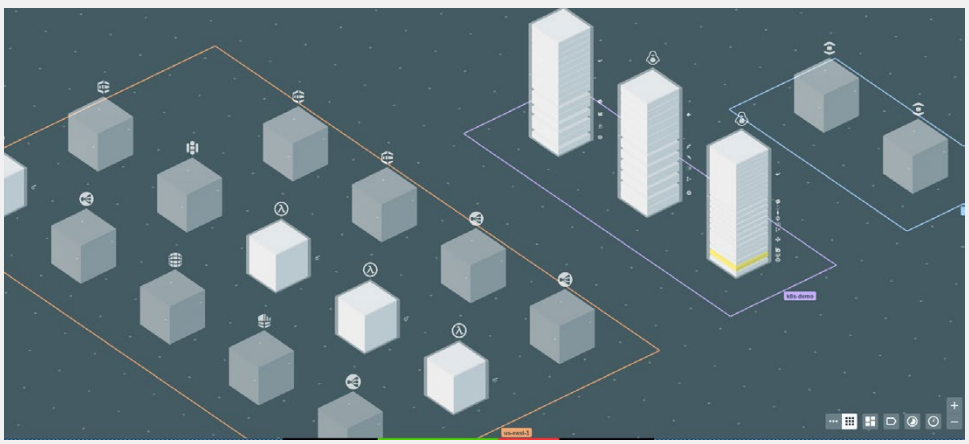
Das Application Monitoring wuchs also zwar mit seinen Anforderungen, wurde jedoch häufig zu einem eigenständigen, zusätzlichen Problem. Häufig überstieg der Aufwand den Nutzen der eingesetzten Technologie.

3. Microservices, Container-Technologien und Cloud-Native-Applikationen

Durch Kubernetes, Microservices, agile Softwareentwicklung und CI-/CD-Ansätze erfuhren Softwarearchitekturen in den letzten Jahren einen weiteren Boost. In rasantem Tempo wuchsen die Applikationslandschaften von Unternehmen weiter, sie steigen nach wie vor exponentiell hinsichtlich ihrer Komplexität. Während IT in der Vergangenheit häufig als Hintergrundprozess verstanden wurde, wird sie von vielen Geschäftsführern unlängst als essentieller Wettbewerbsvorteil erkannt. Release-Zyklen wurden und werden daher immer kürzer, um unternehmensstrategische Gesichtspunkte zu bedienen und Performance Excellence sicherzustellen.

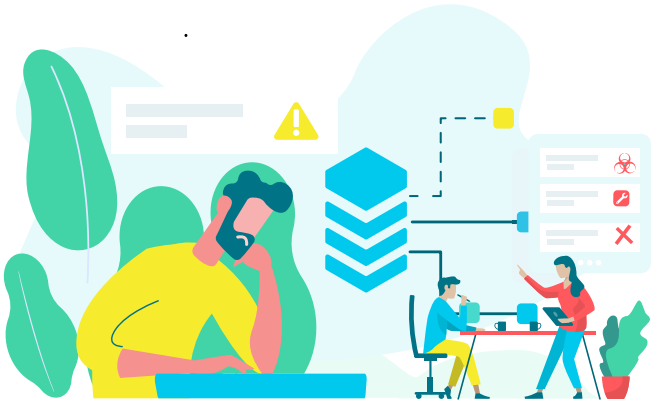
Die Anwendungslandschaft ist dank Cloud und Microservices mittlerweile stark zersplittert und in vielen Unternehmen kaum noch überschaubar. Zu diesem Zweck wurden intelligente Application-Monitoring-Tools auf Basis von *Machine Learning* für das Auffinden von Anomalien im Verhalten von Anwendungen entwickelt. Unterstützt durch künstliche Intelligenz (KI) werden *Root-Cause-Analysen* ermöglicht. Diese zielen auf das automatisierte Aufspüren existierender oder potentieller Performance-Probleme ab und bedingen ein möglichst hundertprozentiges Abfangen aller Calls.

Die neue Generation von APM-Technologien ermöglicht eine sofortige, umfassende Einsicht in alle eingesetzten Systeme ohne aufwendige Konfiguration. Die gesamte Softwarelandschaft wird hierbei sowohl in voller Breite als auch Tiefe analysiert und selbst moderne, chaotische Umgebungen lassen sich transparent darstellen. So werden Performanceprobleme in Sekundenschnelle sichtbar und ihre Relevanz klar dargestellt. Fehlerbehebungen — wie auch Funktionserweiterungen — können derweil im laufenden Betrieb durchgeführt werden und führen zu einer allgemein verbesserten Wartbarkeit des Gesamtsystems. Auch ein effizientes End-User-Monitoring ist nun in umfassender Breite möglich. Je nach Umfang und Beschaffung der IT-Umgebung wird jedoch eine noch tiefergehende, kontextuelle Analysefähigkeit benötigt, die ihrerseits fähig ist, Problemlösungsvorschläge zu unterbreiten.



APM in der Praxis: Wenn die Lösung zum Frust wird

Unter IT-Verantwortlichen herrscht also ein breiter Konsens darüber, dass gegenwärtige IT-Landschaften kaum noch ohne eine systematische Performance-Überwachung zu beherrschen sind. Auf der Suche nach Abhilfe werden in Unternehmen sämtlicher Vertikale daher nicht selten unterschiedliche APM-Ansätze parallel getestet: angefangen bei eigener Entwicklung über Open-Source-Lösungen bis hin zu dedizierter APM-Software. Von Letzteren gibt es in den letzten Jahren einige hoch entwickelte Lösungen (s. Infokasten unter Punkt 3), die Transparenz in die IT-Landschaft bringen können. Nach aufwendiger — und kostenintensiver — Implementierungsphase treten jedoch häufig ganz praktische Probleme auf.



Transparenz: (Auch) ein hausgemachtes Problem

In vielen IT-Abteilungen ist es eine leidige Alltagserfahrung: Entwicklung und Betrieb arbeiten nicht eng genug zusammen, die Absprachen zwischen den Teams sind verbesserungswürdig. Das ist ein ernst zu nehmendes Problem — sind Prozesse und Arbeitsabläufe innerhalb der IT-Abteilung doch genauso wichtig wie die zugehörige Hardware und Infrastruktur. Im Falle von Performance-Schwierigkeiten oder gar Systemausfällen kosten solche Disruptionen häufig wertvolle Zeit und können zu einer Gefahr für das gesamte Geschäft werden.

Abläufe und Kommunikation zu verbessern ist jedoch leichter gesagt als getan. Also sehen viele IT-Leiter die Einführung einer APM-Software als Heilsbringer. Und das keinesfalls zu unrecht: Moderne APM-Tools sind mittlerweile imstande, eine quasi-vollständige Transparenz über den gesamten Technologie-Stack zu gewährleisten und Vernetzungen von Applikationen holistisch darzustellen. Häufig erkennen sie Probleme bereits in der Anbahnungsphase und ermöglichen eine rechtzeitige Intervention durch zielgerichtete Handlungsempfehlungen. Prozesse werden für den Systemarchitekten leicht nachvollziehbar, Flaschenhälse und Hotspots werden schnell erkannt. Werkzeug gekauft. Doch dann?

Der Witterung zum Fraß vorgeworfen

Das Werkzeug ist installiert; der Implementierungsaufwand war enorm, denn das ausgewählte Tool hatte es so erfordert. Die Team-Mitglieder wurden geschult, alle Beteiligten sind zufrieden und sehen das Problem als gelöst an.

Oft zeigt sich dann bald, dass das implementierte Werkzeug doch nicht so wartungsfrei ist, wie man gedacht hatte; die Operations-Mitarbeiter finden im ohnehin straff getakteten Arbeitsalltag nicht genügend Zeit für Wartung, Anpassung und Ausbau des Tools. Die Anwendungslandschaft wächst täglich weiter, die APM-Software misst jedoch die analog dazu erforderlichen, adaptiven Fähigkeiten. Im akuten Problemfall sind die eingewiesenen Mitarbeiter trotz umfassender Schulung von der Komplexität des Werkzeugs oft überfordert oder können die ausgegebenen Meldungen nicht vernünftig interpretieren und sinnvolle Konsequenzen ziehen. Häufig werden Probleme erst gar nicht vom APM-System gemeldet.

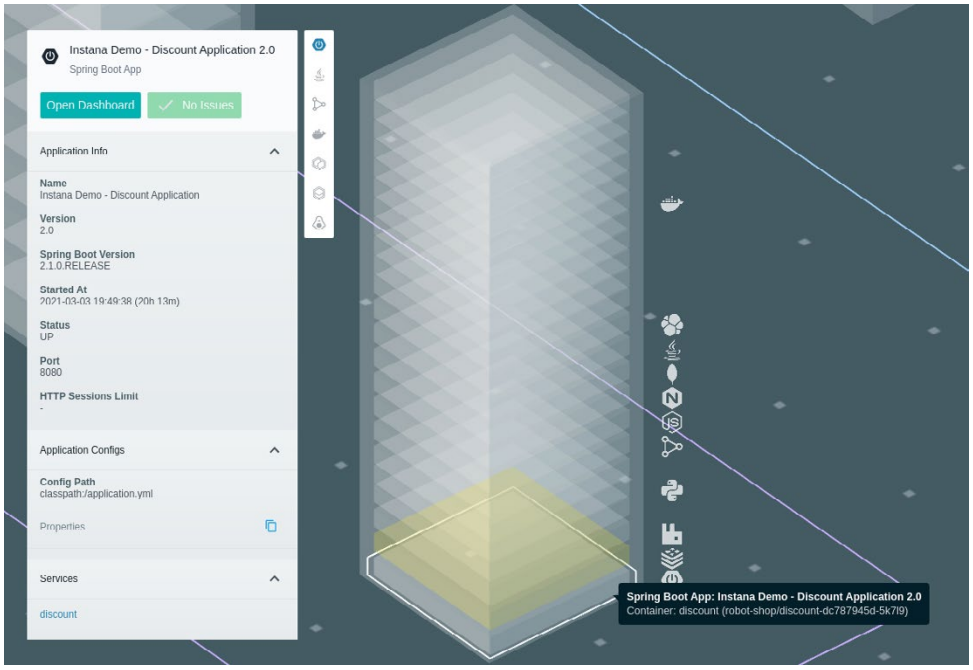
! Die fehlgeschlagene Implementierung eines APM-Tools ist ein häufiges, aber vermeidbares Problem, das mit der Auswahl des richtigen Werkzeugs dezimiert werden kann. Marktgängige Lösungen weisen teilweise große Unterschiede hinsichtlich Adaptivität, Automation und Usability auf, die für den langfristigen Erfolg des Projekts entscheidend sind. Hier ist eine sorgfältige Betrachtung unterschiedlicher Lösungen bzw. eine Beratung durch unabhängige Dienstleister von großem Vorteil.

Spätestens nach einem Jahr resümieren hier viele IT-Leiter, man sei beim Status quo ante angelangt. Das Dilemma liegt nun bei den Entscheidern: Einerseits war die Investition viel zu teuer, um sie einfach wieder zu verwerfen — andererseits verschlingt der Betrieb des Tools beträchtliche Ressourcen, ohne einen merklichen Nutzen zu generieren. Aufgrund der gestiegenen Komplexität der Systemlandschaft ist die Suche nach Performance-Problemen in manchen Fällen sogar aufwendiger als zuvor geworden. So oder so ist das Tool im Folgenden meist zur Legacy verdammt.

Der Grund hierfür ist meist derselbe: Das implementierte Werkzeug erfüllt nicht mehr die Anforderungen der aktuellen Systemlandschaft. Die mittlerweile zu bewerkstelligende Menge an Daten kann von älteren Tools nicht mehr dargestellt werden, sie sind schlichtweg nicht auf die Verarbeitung einer solchen Informationsfülle ausgelegt. Deshalb limitieren viele dieser Tools ihre Aufrufe, die sogenannten *Calls*, per Zufallsprinzip, um deren Fülle verarbeiten zu können. Nach Murphys Gesetz fällt dabei genau der Call durch das Raster, der das Problem aufdecken könnte. Auch verfügen diese Werkzeuge über keinerlei künstliche Intelligenz (KI-Algorithmen) und gehen mit großem Wartungs- und Pflegebedarf einher — eine Mission also, die bereits zum Scheitern verurteilt war, bevor sie begann.

Endlich auf Augenhöhe mit dem Stack

Die gute Nachricht ist: An der Tooling-Front sind in den letzten Jahren einige Quantensprünge vollzogen worden. Während das Monitoring lange Zeit den Entwicklungen der Systemlandschaften stetig hinterherhinkte, lässt die neue APM-Generation berechtigte Hoffnung auf eine nachhaltige Lösung aufkeimen. *Observability* ist die konsequente Weiterentwicklung des APM-Ansatzes, die Management und Monitoring komplexer Anwendungslandschaften auf die nächste Stufe hebt.



Ein schneller Einblick in alle Details des Stacks ist unabdingbar, wenn es zu Störungen oder Ausfällen kommt.

Intelligenteres Monitoring durch Observability

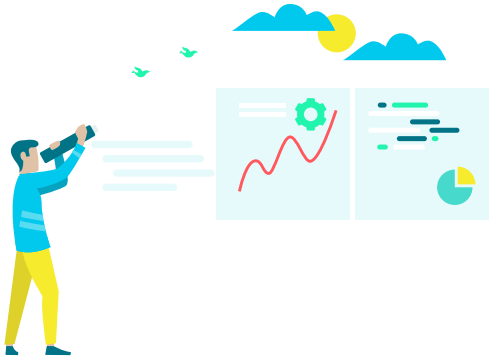
Observability befähigt das Application Monitoring, Kausalzusammenhänge innerhalb des Systems vollumfänglich zu erfassen und besser zu interpretieren. Sie ermöglicht die Untersuchung und das Verstehen des IT-Stacks unter stetiger Berücksichtigung aktueller Anpassungen in der gesamten Anwendungslandschaft. Das Ergebnis sind hochauflösende, feingranulare Diagnosedaten, die Kontext liefern, das Debugging erleichtern und Feedbacks im CI-/CD-Loop verbessern. Dies ist gerade im Hinblick auf die Containerisierung ein wichtiger Aspekt — denn ohne fortlaufende Verbesserungen der Observability können Unternehmen etwa die Vorteile der Cloud-Migration nicht nutzen.

Moderne Systeme ermöglichen sekundengenaue Einblicke in die Performance aller Systemkomponenten.

Mithilfe von Aggregation, Analyse und umfassender Lieferung verschiedenster Telemetrie-Daten wie Logs, Metriken oder Traces lassen sich so intelligente Überwachungsszenarien abbilden, die im Gegensatz zu den vorherigen Generationen von APM-Tools keines aufwendigen Setups bedürfen. Mit Observability ausgestattete Lösungen bieten dabei folgende Vorteile:

- **Systemübergreifender Ansatz:** Die gesamte Applikationslandschaft wird erfasst und in ihrer Ganzheitlichkeit mit allen Interdependenzen betrachtet.
- **Kausalität:** Fehler werden nicht nur auf Applikationsebene betrachtet, sondern auch in den relevanten, systemischen Kontext gesetzt.
- **Intelligenz aus der Cloud:** Cloud-Native-Software befähigt Applikationen, Fehler schnell zu finden und die Ursache zu verstehen.
- **Geschwindigkeit:** Alle Daten werden fokussiert, aufschlussreich und kontextualisiert in Echtzeit aufbereitet.
- **Präzision und Anwenderfokus:** Informationen über technische Fehlfunktionen (z. B. Ausfallzeiten, Fehler und lange Antwortzeiten) werden an vorab definierte Verantwortliche übermittelt.
- **Adaptivität:** Enterprise Observability schlägt die Brücke zwischen Legacy-Applikationen und modernen Hybrid-Cloud-Umgebungen.
- **Automation:** Site Reliability Engineering (SRE) behebt Probleme und automatisiert Operations-Aufgaben, die bislang von Operations-Teams meist manuell ausgeführt werden müssen.
- **Testing und Training:** Chaos Engineering wird zur Stabilitätsprüfung der Anwendungslandschaft durch das Herbeiführen kontrollierter Funktionsunterbrechungen verwendet.

Durch Observability erweiterte APM-Werkzeuge helfen allen Anwendern also, mehrschichtige Architekturen systematisch zu durchdringen. Performance-Einbußen oder Fehler in dynamischen Umgebungen werden von dieser Art Tool nicht nur sehr zuverlässig erfasst, sondern gleich in den relevanten Kontext gesetzt. Dies ist eine beträchtliche Hilfe beim Erfassen der Problemursache und deren nachhaltiger Behebung. Auch hilft es, die Implikationen eines Incidents besser zu verstehen. Neben langfristigen Verbesserungen hinsichtlich Qualität und Stabilität können so auch beträchtliche Einsparungen bei den zusätzlich eingesetzten Personalressourcen erzielt werden.



Quo vadis, APM? – Fazit & Ausblick

Ohne eine leistungsstarke Software zur systematischen Überwachung der Systemlandschaft geht es in den meisten Unternehmen nicht mehr. Viel zu komplex sind die Strukturen geworden, viel zu verteilt liegen die Anwendungen in der Landschaft, viel zu groß sind die Abhängigkeiten. Moderne Microservices-Architekturen und Container-Technologien, wie z. B. Kubernetes oder Docker, dienen heute als Basis großer IT-Landschaften; immer mehr IT-Infrastrukturen und Anwendungen laufen automatisiert in der Cloud. Die Überwachung des Stacks durch klassische Monitoring-Tools ist auch dank CI/CD schlichtweg nicht mehr zu stemmen.

Das klassische APM ist also ein Auslaufmodell, das der systematischen Überwachung verteilter Systeme und insbesondere Cloud-Native- oder Multi-Cloud-Lösungen nicht mehr gerecht werden kann. Die Nutzung einer zukunftsfähigen, automatisierten und skalierbaren APM-Lösung ist daher umso bedeutender für das Aufrechterhalten und den gesunden Betrieb des Gesamtsystems. IT-Entscheider müssen hier umschalten und den Sprung auf die neueste Generation adaptiver, skalierbarer und KI-gestützter APM-Werkzeuge vollziehen. Nur durch eine mitlernende, intelligente Lösung können Unternehmen ihre digitalen USPs erhalten und Effizienz und Umsatz unter Einbeziehung von Wirtschaftlichkeitsaspekten sicherstellen. So werden wieder wertvolle Entwicklungsressourcen frei, die in die Programmierung wertschöpfender Business-Funktionen investiert werden können.

Eine 11-Punkte-Checkliste für die Bewertung eines leistungsstarken APM-Tools finden Sie im Anhang.

Ansprechpartner:

Niko Blättermann
Head of APM
niko.blaettermann@codentric.de

Jetzt Kontakt aufnehmen →

codecentric

Über die codecentric AG

Als Experte für agile und individuelle Software-Entwicklungen ist die codecentric AG seit 2005 Vordenker für innovative Digitallösungen in Deutschland. Dafür kombiniert sie modernste technologische Expertise und ausgefeilte Methoden für und mit ihren Kunden aus allen Wirtschaftszweigen – das Fundament für einen erfolgreichen digitalen Auftritt. codecentrics umfangreiches Projektportfolio reicht dabei von Cloud-Native, Smart Data & AI über IT Integration, Information Security bis hin zu Application Performance Management. codecentric hilft Unternehmen dabei, nachhaltig, schnell und flexibel auf dem Markt zu agieren und befähigt sie, ihre IT-Wertschöpfungskette stabil, sicher und allzeit verfügbar zu machen. Als vertrauensvoller, pragmatischer und erfahrener Partner begleitet die codecentric AG Unternehmen auf dem Weg durch die digitale Transformation und findet mit ihnen zusammen die beste Lösung für ihr Digitalisierungsvorhaben. Nicht zuletzt deswegen ist codecentric von Instana zum ‚EMEA Partner of the Year 2020' gekürt worden.

11-Punkte-Checkliste:

So finden Sie das beste APM-Werkzeug

Die folgenden elf Aspekte sollten berücksichtigt werden, wenn Sie mithilfe einer APM-Lösung Ihre verteilte Applikationslandschaft überwachen möchten:

1. **Schneller Proof of Concept (PoC):** Die Anbindung sollte nach wenigen Stunden abgeschlossen und erste Ergebnisse direkt sichtbar sein. Wenn es bereits am Anfang umständlich und aufwendig wird, sollten Sie lieber die Finger davon lassen.
2. **Vollständige Daten:** Achten Sie darauf, dass Performance-Daten vollständig (100 % aller Calls!) erfasst werden und nicht auf Sampling-Verfahren zurückgegriffen wird.
3. **Echtzeitanalyse**: Über automatische, fortlaufende Systemüberwachung (CI/CD) und Modellierung sollte jederzeit ein aktuelles Bild der Applikationsstrukturen und ihrer Abhängigkeiten abrufbar sein.
4. **Übersichtlichkeit und Detailtreue**: Das Werkzeug muss den Spagat zwischen systemübergreifender Darstellung und Darstellungstiefe jederzeit und auf Abruf meistern können. Ein Zoom-in in jede beliebige Komponente des Systems und seine Schnittstellen sollte immer möglich sein. Anwendungen und ihre Abhängigkeiten sollten dabei grafisch übersichtlich und leicht zu erfassen dargestellt werden.
5. **Lernfähigkeit durch KI:** Die Lösung passt sich automatisch und ohne jegliche Konfiguration den Änderungen in der Systemlandschaft (wie Cloud-Native-Anwendungen, Containern oder Microservices) an. Eine manuelle Justierung darf nicht mehr erforderlich sein.
6. **KI-gestützte Alerts und Forecasts:** Das lernende System erfasst relevante Auswirkungen und sagt potentielle Verfügbarkeitsausfälle voraus.
7. **Intelligente Fehlersuche:** Die Software unterstützt die Lösungsfindung proaktiv und lernt im Überwachungsprozess kontinuierlich mit.
8. **Einfache Handhabung:** Das APM-Tool sollte weitestgehend selbsterklärend und intuitiv zu bedienen sein. Anwender jeden Levels sollten Alerts und Benachrichtigungen problemlos verstehen können.
9. **Messbare Verbesserung abteilungsinterner Kennzahlen:** Merkliche Reduzierung der Mean Time To Repair (MTTR), Freiwerden von Entwicklungsressourcen, verbesserte Einhaltung von SLAs: Setzen Sie sich konkrete KPIs, um den Erfolg des Tools zu messen.
10. **Transparentes Lizenzmodell:** Automatisierte Kernfunktionen wie das Monitoring von Kubernetes, End-to-End-Monitoring, Serverless, Cloud und Infrastructure, AIOps, Alerting, Analytics oder Logging sollten in der Lizenz inkludiert sein, ebenso die Kosten für alle Traces, Datenbanken, User und Storage. Das Modell sollte einfach zu verstehen sein, damit Sie sich nicht um unvorhergesehene Kosten sorgen müssen.
11. **Rollen-Management:** Jeder Mitarbeiter sollte die für ihn relevanten Daten auf seine Bedürfnisse abgestimmt aufbereitet und dargestellt erhalten.