White Paper

USU-Expertise im IT-Event-Management

Best Practices für ein leistungsfähiges IT-Monitoring

Inhalt

Einleitung 3

Die Aufgaben des IT-Event-Managements 4

Metriken und Events 5

Best Practices für das IT-Event-Management 6

Die 3 Prozessschritte des Event-Managements 6

Messen 7

Bewerten 10

Reagieren 14

Fazit 14

Einleitung

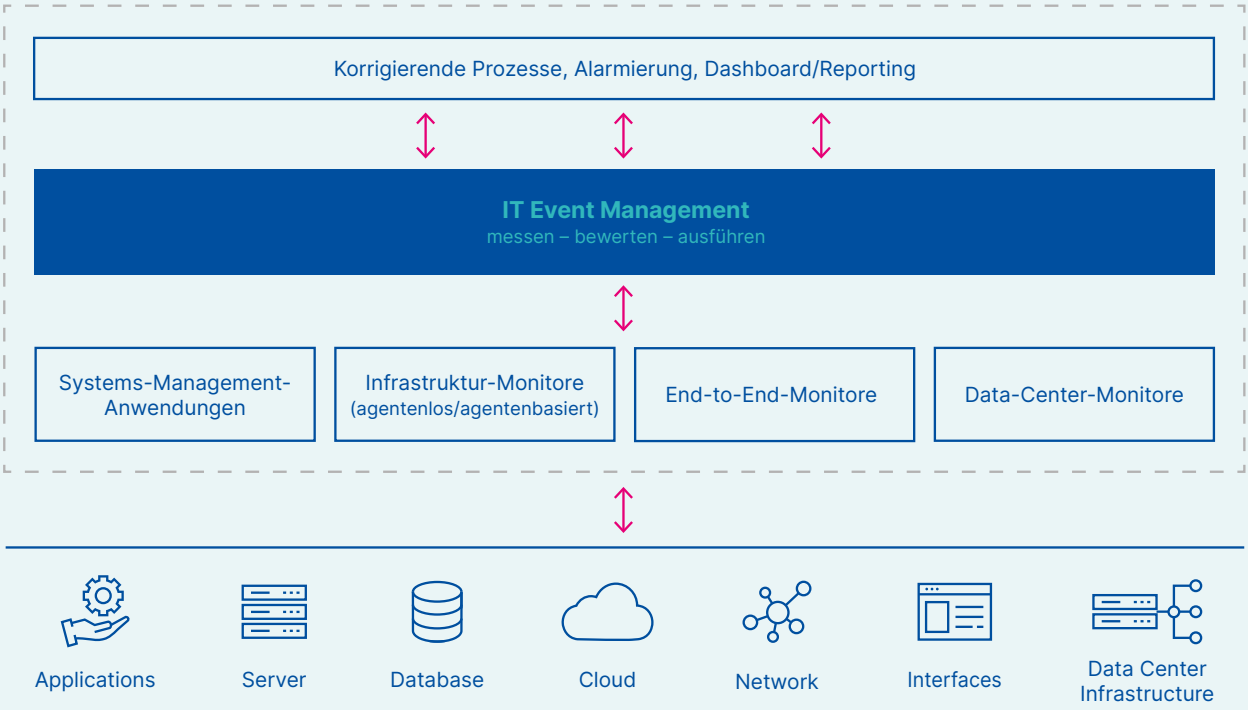
Zur ganzheitlichen Überwachung der IT-Infrastruktur und der darauf laufenden IT und Business Services kommen unterschiedliche Informationsquellen zum Einsatz. Jede dieser Quellen liefert eine Menge von Daten, die Zustandsänderungen (Events) der Infrastruktur identifizieren. Während des IT-Betriebs entsteht ein unablässiger Strom von Events, der nur mit Hilfe eines automatisch ablaufenden Event Managements effizient ausgewertet werden kann. Ziel dieses Event Managements ist es, im Datenstrom relevante Warnsignale frühzeitig und zuverlässig zu erkennen und somit durch proaktive, korrigierende Maßnahmen Probleme im IT-Betrieb zu verhindern.

Ein effizientes IT-Event-Management ist somit eine der entscheidenden Voraussetzung für einen reibungslosen IT-Betrieb. Wie das Event-Management funktioniert und worauf Sie bei der Implementierung achten sollen, erfahren Sie in diesem Whitepaper.

Die Aufgaben des IT-Event-Managements

Das Event-Management ist eine der tragenden Säulen beim IT-Monitoring. Es wertet die Daten der zahlreichen Monitoring-Quellen kontinuierlich aus, erkennt problematische Situationen und stößt behebende Maßnahmen an.

Abbildung 1: Event Management der USU als Herzstück des Monitorings



Ein Beispiel dafür ist, dass die CPU-Last eines Servers seit mehreren Minuten auf 95 Prozent steht. Die CPU-Last eines anderen Servers ist für wenige Sekunden auf 96 Prozent gestiegen, danach aber wieder auf 50 Prozent gefallen ist. Beide Statusänderungen lösen Events aus, aber nur ein Status ist aktuell relevant für den Support. Ein Event-Management-System erkennt die wichtigen Events und wird nur dann aktiv.

Das Event Management kann Prozesse anstoßen, die automatisch bestimmte Probleme beheben oder diesen vorbeugen können. Es kann aber auch Administratoren alarmieren, wenn eine manuelle Bearbeitung notwendig ist. In jedem Fall verhindert das Event Management, dass Administratoren Fehlalarme bearbeiten oder in einer Flut von Events die tatsächlichen Probleme nicht erkennen.

Beispiel: Im Netzwerk ist ein Mail-Server ausgefallen. Da dieser Server aber nur als einer von drei Knoten in

einem Mail-Cluster eingebunden ist, bleibt der tatsächliche Mail-Service ungestört. Der Serviceverantwortliche muss daher nicht unbedingt sofort benachrichtigt werden. Es reicht aus, wenn er bei nächster Gelegenheit eine Warnmeldung im Dashboard sieht und die nötigen Schritte einleitet.

Fällt während der täglichen Arbeitszeiten ein Service aus, sollte ein Incident generiert werden, da die Produktivität der Mitarbeiter eingeschränkt wird. Wenn der gleiche Ausfall außerhalb der Geschäftszeiten erkannt wird, ist es nicht zwingend notwendig, einen Incident zu erstellen. Das sind nur zwei Beispiele, wie ein Event Management proaktiv Bewertungen ausführt und Administratoren einbindet.

Metriken und Events

Der Begriff „Metrik“ kommt aus dem Griechischen und bedeutet Messung. Monitore erfassen kontinuierlich Metriken wie z. B. die CPU-Auslastung, die Antwortzeiten einer Anwendung oder die Verfügbarkeit eines Servers.

Statusänderungen in den Metriken können Events auslösen. Wenn zum Beispiel ein überwachter Webdienst über längere Zeit eine stark ansteigende Anzahl an Abfragen pro Minuten verarbeiten muss und gleichzeitig die Auslastung der CPU und des Arbeitsspeichers der verantwortlichen Server stark ansteigen, bis hin zur Überlastung, löst dies mehrere Events aus:

- Die Abfragen pro Minuten steigen über den definierten Schwellwert
- Die CPU-Last steigt über den definierten Schwellwert
- Die Auslastung des Arbeitsspeichers steigt über den definierten Schwellwert

Best Practices für das IT-Event-Management

Die 3 Prozessschritte des Event-Managements

Damit das Event-Management seine Aufgaben zuverlässig erfüllen kann, spielen drei Prozessschritte eine wichtige Rolle:

01 | Messen – Sowohl Metriken als auch Events werden kontinuierlich erfasst

02 | Bewerten – Die Messwerte werden analysiert und bewertet

03 | Reagieren – Abhängig vom Bewertungsergebnis werden Aktionen gestartet

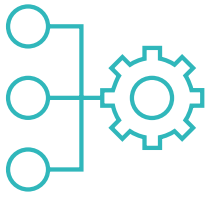
Ein leistungsfähiges Event-Management ermöglicht die getrennte und unabhängige Konfiguration dieser drei Schritte. So können beispielsweise neue Infrastrukturen in die Messung mit eingebunden werden, ohne die Bewertungs- und Ausführungsprozesse ändern zu müssen. Oder es können neue Alarmierungswege implementiert werden, ohne dass diese Auswirkungen auf die Bewertung haben.

Bei der Implementierung des ersten Prozessschritts, also dem „Messen“, sollte man zu Beginn eher zurückhaltend sein. Es droht die Gefahr, dass man viele Daten erfasst, speichert und bewertet, die für eine effektive Überwachung gar nicht notwendig sind. Also lieber mit weniger Daten beginnen, und dann bei Bedarf weitere Daten in die Überwachung übernehmen.

Für die Implementierung des letzten Prozessschritts, also das „Reagieren“, hat sich in der Praxis ein Stufenmodell bewährt, bei dem Schritt-für-Schritt Standardvorgehensweisen durch spezifischere Vorgehensweisen ersetzt werden. Wichtig dabei ist, dass Event-Attribute grundsätzlich Standardwerte, objektspezifische Werte oder Event-spezifische Werte annehmen können. Die Zuweisung dieser Werte läuft dann über besagtes Stufenmodell ab.

Beispiel: in der ersten Projektstufe wird als Alarmierungsziel in allen Events immer ein zentrales Administratoren-Team als Standardwert eingesetzt. In der zweiten Projektstufe wird dieser Wert durch Objektspezifische Werte überschrieben. Abhängig vom Typ der betroffenen Infrastruktur wird dann als Alarmierungsziel z. B. das Windows-Server-Team oder das Netzwerk-Team definiert. In der letzten Projektstufe werden objektspezifische Werte nach und nach durch Event-spezifische Werte ersetzt, das Alarmierungsziel richtet sich dann nach dem Typ der Events.

Auf diese Weise ist sichergestellt, dass ein Event-Attribut wie beispielsweise das Alarmierungsziel in jedem Fall einen gültigen Wert besitzt und dennoch größtmögliche Flexibilität vorhanden ist, um das Monitoring nach und nach an spezifische Anforderungen anzupassen.



Messen

Für das Erfassen von Metriken und Events gibt es in der Praxis eine Vielzahl von Quellsystemen.

Systems-Management-Anwendungen

Zur übergreifenden Administration von Infrastrukturen desselben Typs werden häufig Systems-Management-Anwendungen eingesetzt. Beispiele hierfür sind:

- Administration von Virtualisierungsumgebungen durch VMWare vSphere oder Microsoft Hyper-V
- Administration von Datenbanken durch Oracle Enterprise Manager oder Microsoft MQL Server Manager
- Administration von Cloud-Infrastrukturen durch die Portallösungen der Cloud Provider wie Amazon AWS oder Microsoft Azure
- Administration von Container-Umgebungen durch Kubernetes oder Openshift

Diese Systems-Management-Anwendungen haben eigene Überwachungsfunktionen und sind damit eine wichtige Monitoring-Quelle. Durch spezifische Konnektoren, die die proprietären APIs der Hersteller kapseln, greift das IT-Event-Management auf diese Systems-Management-Anwendungen zu. Über diese Schnittstellen werden Messwerte (Metriken) und auch Warn- oder Fehlerzustände (Events) ausgelesen. Nach Behebung eines Problems kann das IT-Event-Management auch eine OK-Nachricht zurücksenden, damit der Warn- oder Fehlerzustand von der Konsole der Systems-Management-Anwendung verschwindet.

Infrastruktur-Monitore

Grundsätzlich gibt es eine Vielzahl unterschiedlicher Monitoring-Tools zur Überwachung einzelner Infrastruktur-Komponenten. Eine leistungsfähige, übergeordnete Monitoring-Suite sollte grundsätzlich eigene Infrastruktur-Monitore mitbringen als auch bereits vorhandene Monitore wie z. B. Nagios, SCOM oder WhatsUpGold einbinden können.

Agentenlose Monitore

Agentenlose Monitore greifen über Standardschnittstellen von außen auf die zu überwachenden Infrastrukturen zu. Mittels ICMP-Ping wird erkannt, ob das Zielsystem überhaupt läuft. Per SNMP und CIM/WMI können Zustandsinformationen wie Festplatten- oder RAM-Auslastung aus Servern und Datenbanksystemen ausgelesen werden.

Dieses minimal-invasive Messverfahren ist deutlich einfacher zu implementieren als ein agenten-basiertes. Auf der zu überwachenden Hardware muss keine Client-Software installiert werden, das Hinzufügen oder Ändern der zu überwachenden Infrastruktur ist schnell möglich. Dafür ist das agentenlose Monitoring aber auch deutlich weniger leistungsfähig. Außerdem müssen für jede genutzte Schnittstelle die entsprechenden Ports geöffnet werden.

Agentenbasierte Monitore

Durch Installation von Agenten auf den Servern für Applikationen und Datenbanken lassen sich diese deutlich umfassender überwachen. Dieses Verfahren ist zwar aufwändiger als das der agentenlosen Überwachung, allerdings erzielt man auch wesentliche Vorteile:

01 | **Transaktionssichere Überwachung:** auch während eines Netzwerksausfalls sammelt der Agent weiterhin Überwachungsdaten und puffert diese so lange, bis die Netzwerksverbindung wieder steht und danach alles an das IT-Event-Management übertragen werden kann. Bei agentenlosen Systemen kann nachträglich keine Aussage mehr über den Zustand der Infrastrukturen getroffen werden. Transaktionssicherheit lässt sich daher nur durch Agenten erreichen.

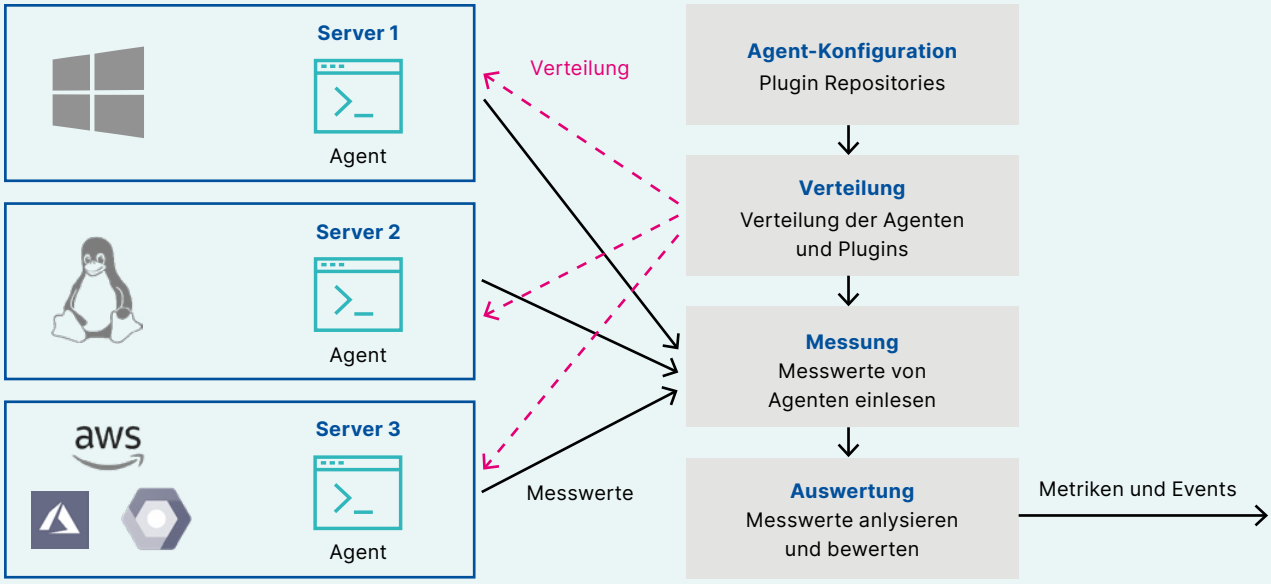
02 | **Tiefergehende Informationen:** nachdem der Agent direkt auf dem Zielsystem läuft und auch in der Lage ist, Skripte zu starten, kann er jede gewünschte Information liefern. Das können Warnungen oder Fehlermeldungen aus Log-Files sein. Oder die Bestätigung der korrekten und performanten Funktion einer Datenbank durch Ausführen von SQL-Skripten oder einer Web-Applikation durch Analyse des zurückgelieferten HTML-Codes.

03 | **Entlastung des Netzwerks:** Der Agent sammelt die benötigten Informationen auf dem Server, analysiert sie und baut dann eine Verbindung zum Monitoring-

System auf, um nur die relevanten Daten zu senden. Vor allem in großen Netzwerken kann durch gut geplanten Einsatz der Agenten die Last im Netzwerk signifikant reduziert und die Sicherheit der Kommunikation erhöht werden. Dadurch reduziert sich gleichzeitig der Aufwand für die Netzwerkkonfiguration und die Anzahl offener Firewall-Ports.

Wichtig für den Einsatz von Agenten ist es, dass diese das System so wenig wie möglich belasten und so einfach wie möglich installiert werden können, idealerweise automatisiert über das Netzwerk.

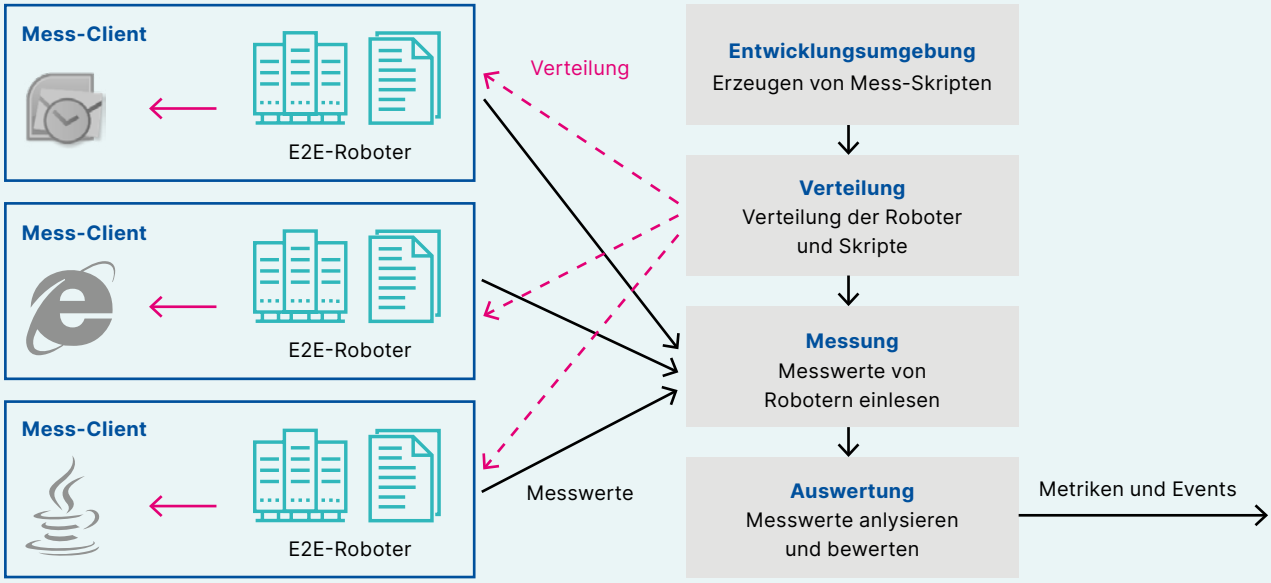
Abbildung 2: Agenten-basiertes Monitoring der USU



E2E-Monitore

End-to-End-Monitore (E2E) werden zur Überwachung von Applikationen eingesetzt. Anders als die agentenlosen oder agenten-basierte Monitore greifen sie nicht direkt auf die Infrastrukturen zu. Stattdessen verwenden sie dieselben Schnittstellen, die die Anwender der Applikation auch nutzen: das sind also HTTP(S) für webbasierte Anwendungen und die Benutzeroberfläche nativ installierter Clients. Zur Überwachung bilden sie das Verhalten eines interaktiven Benutzers nach. Wie ein echter Anwender meldet sich ein E2E-Roboter an der zu überwachenden Anwendung an und durchläuft ein definiertes Testprotokoll. Durch Analyse der Antwort kann der End-to-End-Monitor die korrekte Funktion der Anwendung feststellen und Metriken wie Verarbeitungsgeschwindigkeit oder Antwortzeiten messen.

Abbildung 3: End-to-End Monitoring der USU



E2E-Monitore sind wichtige Hilfsmittel zur Messung von Anwender-bezogenen Metriken. Legt ein Service Level Agreement beispielsweise maximal zulässige Antwortzeiten der Benutzeroberfläche einer Applikation fest, oder eine garantierte Bearbeitungszeit bestimmter Aufgaben, kann dies zuverlässig nur mit E2E-Monitoren gemessen werden.

Damit sich der administrative Aufwand für das E2E-Monitoring in Grenzen hält, sind folgende Eigenschaften wichtig:

- einfaches Erweitern oder Ändern der Mess-Skripte
- zentrale Administration und Verteilung der E2E-Roboter auf den Mess-Clients
- zentrales Task-Scheduling für das automatisierte Ausführen der Mess-Skripte
- zuverlässiges Überwachen, ob die End-User-Simulation erfolgreich durchlaufen werden konnte

Gerade der letzte Punkt ist wichtig, denn nur so kann erkannt werden, ob die zu überwachende Applikation geändert wurde und deshalb die Mess-Skripte angepasst werden muss.

Data-Center-Monitore

Zu einer ganzheitlichen Überwachung der IT gehört auch die Überwachung von Gebäuden und Räumen des Rechenzentrums. Das ermöglicht die frühzeitige Erkennung der sinkenden Leistung der Klimatisierung, die drohende Überlastung der Stromschienen oder Störungen an der USV-Anlage.

Als Messquellen stehen Temperatursensoren, Netzanalysatoren und RCM-Geräte zur Verfügung. Alternativ können die Daten auch indirekt von vorgelagerten Systemen wie einer Gebäudeleittechnik oder einer Brandmeldezentrale abgefragt werden.

Bewerten

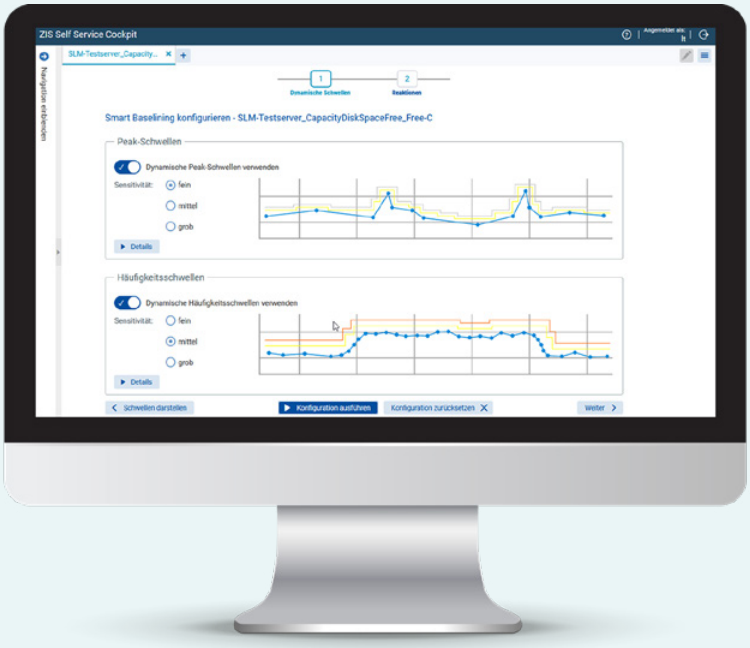
Durch die verschiedenen Messquellen erhält das Monitoring-System einen unablässigen Strom von Metriken und Events. Diese müssen nun analysiert und bewertet werden, um daraus zuverlässige Aussagen über den Gesundheitszustand von Systemen und Anwendungen zu erstellen und ggf. proaktive, korrigierende Prozessschritte zu starten.

KI-basierte Anomalie-Erkennung

Zur Überwachung von Metriken wie z. B. der CPU-Auslastung eines Servers werden heute oft noch statische Schwellwerte eingesetzt. Ein Über- oder Unterschreiten führt automatisch zur Benachrichtigung der zuständigen Administratoren, die dann in einer nachfolgenden Analyse feststellen, ob tatsächlich ein Problem vorliegt, welches abstellende Maßnahmen erfordert. Das Dilemma bei dieser Methode: werden die Schwellwerte zu eng gesetzt, führt dies zu vielen manuellen Analysen, die häufig feststellen, dass gar kein Problem vorliegt. Werden die Schwellwerte zu weit gefasst, bleiben möglicherweise Probleme unerkannt, die später eine Verletzung des Service Level Agreements oder einen Serviceausfall verursachen können.

In dynamischen Infrastruktur-Umgebungen ist die optimale Festlegung eines statischen Schwellwertes kaum möglich. Deshalb sollte die Überwachung von Metriken heute KI-basiert erfolgen. Der KI-Algorithmus analysiert mittels Machine Learning kontinuierlich Messdaten wie z. B. E2E-Messungen, Server- und Netzwerk-Performance oder Daten aus Log-Einträgen. Er analysiert die historischen Verläufe der Metriken und identifiziert die Muster, die mit Störfällen aus der Vergangenheit korrelieren. Die KI lernt somit, welche Zeitverläufe der gemessenen Metriken normal sind und welche auf eine Anomalie hinweisen, die zu einem Störfall führen wird. Dieses Wissen wird dann im laufenden Betrieb angewandt. Nur im Falle einer Anomalie in den aktuellen Messreihen wird tatsächlich ein Event zur Weiterverarbeitung erzeugt.

Abbildung 4: KI-basierte Anomalie-Erkennung der USU



Vertiefen Sie Ihr Wissen:

Die KI-basierte Anomalie-Erkennung der USU im Detail – Jetzt entdecken!

Dieses Verfahren wird „Smart Baselining“ genannt. Das Trainieren der Mustererkennung auf Basis historischer Daten findet dabei fortlaufend statt. Die KI funktioniert also wie eine dynamische Schwellwertanpassung, die sich ständig auf Basis historischer Daten optimiert. Unnötige Fehlalarme werden vermieden, „echte“ Unregelmäßigkeiten dagegen schnell erkannt. Dadurch steigt die Monitoring-Qualität, bei geringerem Arbeitsaufwand.

Regelbasierte Event-Korrelation

Die Event-Korrelation hat die Aufgabe, die zahlreichen Events aus den unterschiedlichen Messquellen zu analysieren, zu bewerten, Probleme zu diagnostizieren und letztlich Aktionen zu deren Behebung zu veranlassen.

Events dürfen dabei nicht einzeln und isoliert betrachtet werden, denn häufig resultiert ein und derselbe Fehler in mehreren aufeinanderfolgenden, getrennten Events. Der Ausfall eines Business Service beispielsweise kann sich durch Events aus dem E2E-Monitoring, dem Applikations-Monitoring und dem Server-Monitoring zeigen. Events müssen also miteinander korreliert werden, um redundanten Information herauszufiltern und die eigentliche Fehlerursache lokalisieren zu können. Bei der Korrelation sind grundsätzlich zwei Zusammenhänge zu berücksichtigen:

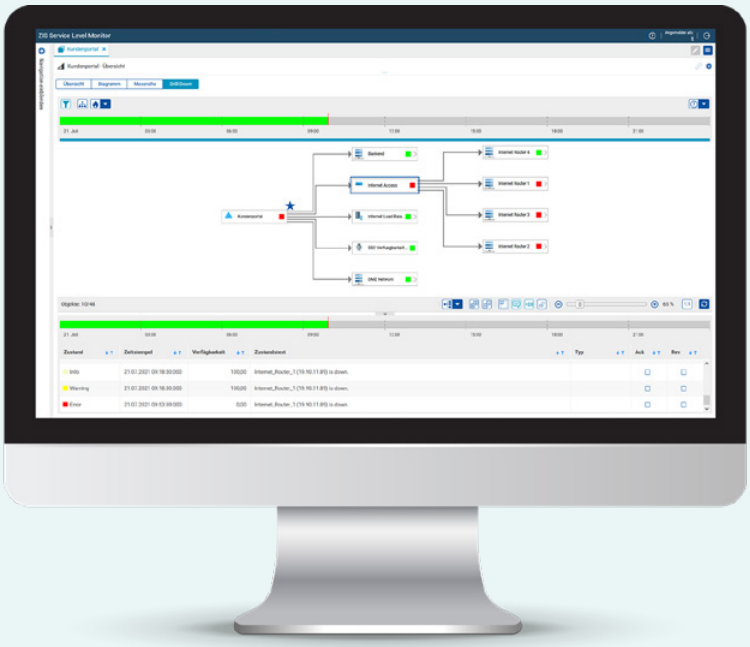
01 | Zeitliche Korrelation

Events werden miteinander korreliert, wenn sie innerhalb eines bestimmten Zeitfensters gehäuft auftreten und somit auf dieselbe Fehlerursache hinweisen. So deutet z. B. ein mehrmaliges Überschreiten des Schwellwertes für die CPU-Auslastung innerhalb der letzten 5 Minuten auf einen Kapazitätsengpass hin.

02 | Topologische Korrelation

Events werden miteinander korreliert, wenn sie von Infrastrukturelementen erzeugt werden, die alle zu ein und demselben Business Service gehören. Die Event-Korrelation muss dazu Kenntnis über die Servicestrukturen (Topologie) der Business Services haben. Servicestrukturen bestehen häufig aus logischen und physikalischen Servicekomponenten. Physikalische Servicekomponenten sind die Infrastrukturelemente auf der untersten Ebene wie Server oder Datenbanken. Logische Servicekomponenten sind übergeordnete Sammelobjekte, die jeweils die Objekte der darunterliegenden Ebene bündeln.

Abbildung 5: Servicestruktur



Da die logischen Servicekomponenten nicht durch physikalische Discovery-Verfahren wie z. B. Netzwerks-Scans erkannt werden können, sollten sie entweder direkt im Monitoring-Tool modelliert oder aus einer externen Configuration Management Database (CMDB) automatisch eingelesen werden können.

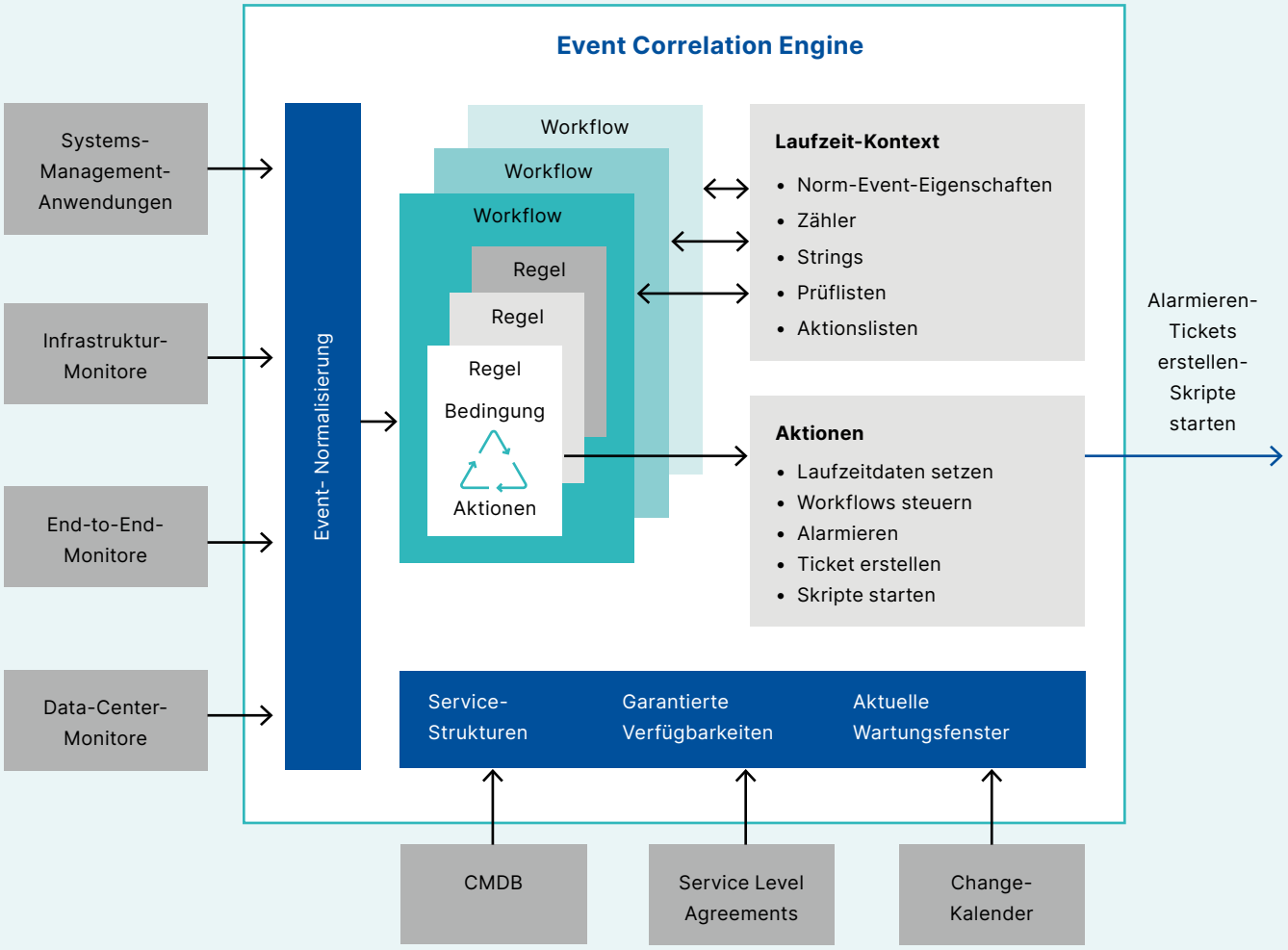
Damit das Regelwerk für die Event-Korrelation flexibel angepasst werden kann, sollte es aus drei Bausteinen bestehen:

01 | Workflows: sie werden von den Events gestartet und bestimmen den Ablauf von Prüfschritten und resultierenden Aktionen.

02 | Regeln: sie enthalten die zu prüfenden Bedingungen für einen Prüfschritt.

03 | Aktionen: sie definieren, was je nach Prüfergebnis zu tun ist.

Abbildung 6: Regelbasierte Event-Korrelation der USU



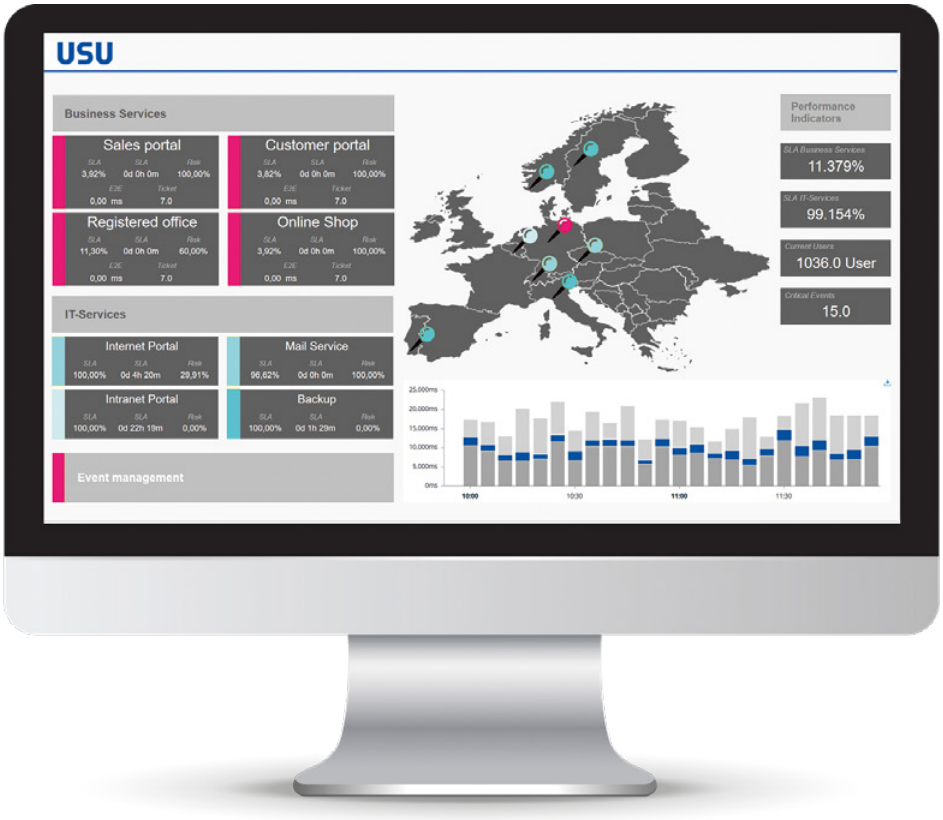
Für alle gängigen Überwachungsaufgaben sollte ein Monitoring-Tool bereits ein komplettes Set an vordefinierten und praxiserprobten Workflows mitbringen, wie zum Beispiel zur Überwachung von Windows-Server, Linux-Server, Microsoft Exchange, Microsoft SharePoint, VMware vSphere, Kubernetes, Openshift oder Cloud-Services der etablierten Anbieter. Mit Hilfe des flexiblen Regelwerks werden diese dann um kundenindividuelle Überwachungsaufgaben erweitert.

Bei der Bewertung von Events spielt auch eine große Rolle, ob und welche Service Level Agreements vereinbart wurden. So muss erst dann ein Alarm ausgelöst werden, wenn z. B. durch wiederholte Ausfälle sich die Ausfallszeiten kontinuierlich addieren und letztlich die zugesagte Verfügbarkeit eines Business Service bedroht wird. Ebenso muss berücksichtigt werden, welche Infrastrukturen aktuell Wartungsarbeiten unterzogen sind und deshalb bei Fehlermeldungen durch die Monitore eine Alarmierung unterbunden werden muss.

Sowohl Service Level Agreements als auch Wartungsintervalle sollten im Monitoring-Tool konfiguriert als auch über Schnittstellen von IT-Service-Management-Tools eingelesen werden können.

Dashboards und Reports

Sowohl für Administratoren als auch für das Management sollte das Monitoring-Tool umfangreiche Dashboards und Reports zur Verfügung stellen. Wichtig hierfür sind das revisionssichere Speichern der gemessenen Daten auch über einen längeren Zeitraum hinweg und die Konfiguration zielgruppenspezifischer Ansichten.



Reagieren

Hat die Event-Korrelation ein Problem diagnostiziert, müssen korrigierende Maßnahmen durchgeführt werden.

Skripte starten

Skripte bieten eine einfache Möglichkeit, grundsätzlich jede Art von korrigierenden Prozessen automatisch zu starten. So können z. B. abhängig von der CPU-Auslastung automatisch Skalierungsprozesse gestartet werden, um zusätzliche Serverkapazitäten auf- und auch wieder abzubauen.

Ticket erstellen

Soll ein diagnostizierter Störfall im Rahmen des Incident-Managementprozesses behandelt werden, muss ein Ticket im ITSM/Helpdesk-Tool durch Aufruf einer Web-API erstellt werden. Wichtig ist dann allerdings auch der Rückweg: nach Behebung der Störung muss der Fehlerzustand im Monitoring-System und ggf. auch im Quellsystem (Systems-Management-System) wieder zurückgesetzt werden. Für diesen Rückweg muss das Monitoring-Tool eine Web-API zur Verfügung stellen, die aus dem Ticket-Prozess aufgerufen werden kann.

Alarmieren

Im Ernstfall müssen Personen unmittelbar alarmiert werden, damit Rettungsaktionen unverzüglich erfolgen können. Wichtig ist die Möglichkeit, flexibel zu konfigurieren, wer, wann, über welchen Kanal und über welche Events informiert werden soll. Quittierungsfunktionen stellen sicher, dass der Alarm auch tatsächlich ankam. Eskalationen müssen gestartet werden, wenn die Quittierung nicht rechtzeitig erfolgt.

Ebenfalls wichtig ist die Nutzbarkeit mehrerer Alarmierungskanäle mit möglichst hoher Verfügbarkeit. Dazu kann man neben E-Mails und MS-Teams-Nachrichten auch SMS oder Alarmierungs-Apps für mobile Endgeräte nutzen.

Fazit

In komplexen, heterogenen IT-Umgebungen kann eine leistungsfähige Überwachung nur mit Hilfe eines automatisch ablaufenden Event Managements gewährleistet werden. Die Kombination aus KI-basierten Algorithmen und flexiblen Regelwerken stellt dabei sicher, dass aus der Menge der Monitoring-Informationen die relevanten Events automatisch ausgesiebt und nur diese zur weiteren Verarbeitung an den Support gemeldet werden. Und das führt letztlich zu einem reibungslosen IT-Betrieb mit minimalem Arbeitsaufwand.

Über USU

USU ist Deutschlands Nr. 1 für IT-Monitoring-Lösungen. Das umfangreiches Leistungsspektrum im Bereich IT Monitoring erstreckt sich über die gesamte Entwicklung und Implementierung von Monitoring-Lösungen, den Know-how-Transfer in die jeweiligen IT-Abteilungen sowie den erstklassigen Support und die zuverlässige Wartung der Software. Mit einer unübertroffenen Expertise und einer langjährigen Erfolgsgeschichte ist die USU in der Lage, auch individuelle Kundenanforderungen zu berücksichtigen und maßgeschneiderte Lösungen anzubieten.

Benötigen Sie weitergehende Informationen, eine Live-Demo oder haben Sie Fragen? Die meisten Fragen lassen sich im direkten Kontakt am besten klären. Wir freuen uns darauf, Ihre Fragen und Wünsche telefonisch zu beantworten. **Jetzt einen Termin vereinbaren.**

Melisa Mujic
ITM Community Developer USU Solution IT Monitoring

USU-202401