



Search. Observe. Protect.

# Einführung in APM

Wie, was und warum

[elastic.co/de](https://elastic.co/de)

# Inhaltsverzeichnis

Warum brauchen wir Monitoring der Anwendungsleistung (Application Performance Monitoring, APM)?	3
Unternehmen sind auf Anwendungen angewiesen	3
Leistungsprobleme bedeuten Ertragseinbußen	4
Moderne, cloudnative Apps sind komplex	4
Was genau ist APM?	5
End-to-End-Transparenz	5
Latenz und Fehler sichtbar machen	6
APM über die Produktion hinaus	6
Wie funktioniert APM?	7
APM-Agents	7
Instrumentierung und Konfiguration	8
Analyse	8
Wichtige Begriffe	9
Verteiltes Tracing	9
Spans	10
Transaktionen	10
Traces	11
Real User Monitoring (RUM)	11
Auswahl eines APM-Tools	12
Technische Fähigkeiten	12
Nutzerfreundlichkeit	13
Deployment-Optionen	13
Unterstützung für offene Standards und offene Daten	13
Architektur und Skalierbarkeit	14
Security	14
Fähigkeiten, die über APM hinausgehen	15
Preise	15
Über Elastic	16

# Warum brauchen wir Monitoring der Anwendungsleistung (Application Performance Monitoring, APM)?

**Kurzer Hinweis:** In diesem Leitfaden besprechen wir das [Monitoring der Anwendungsleistung \(APM\)](#) als einen der Grundpfeiler der [Observability](#). Zusammen mit Logs und Metriken spielt APM eine entscheidende Rolle bei der Erstellung von Systemen mit Observability.

## Unternehmen sind auf Anwendungen angewiesen

Anwendungen sind der öffentliche Auftritt moderner Organisationen. Sie sind unsere Schnittstelle zu Produkten und Diensten, egal ob es sich um eine E-Commerce-Site, eine Mitfahr-App oder um die verschiedenen Zusammenarbeits- und Produktivitäts-Tools geht, die wir tagtäglich verwenden. Wenn wir ein Paar Kopfhörer in unseren digitalen Einkaufskorb legen oder die perfekte Playlist für eine Dinnerparty zusammenstellen, interagieren wir mit – und bilden unsere Meinung über – Anwendungen. Markenwahrnehmung, Treue und Bewertungen basieren auf diesen Erlebnissen.

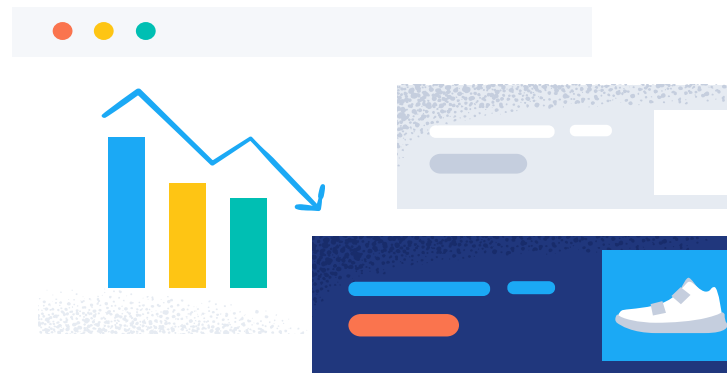


# Leistungsprobleme bedeuten Ertragseinbußen

Denken Sie an die Anwendungen, mit denen Sie arbeiten, oder an Ihre persönlichen Geräte zuhause. Was fällt Ihnen ein? Wie benutzerfreundlich sind diese Apps und Geräte? Wie schnell reagieren sie? Wie häufig treten Fehler auf? Wirken sich die Antworten auf diese Fragen auf Ihre Wahrnehmung der Unternehmen aus, von denen diese Angebote stammen?

Wenn Anwendungen nicht wie erwartet funktionieren, wirkt sich dies direkt auf das Geschäft aus. Erträge und das Vertrauen der Verbraucher stehen auf dem Spiel.

Wenn ein digitaler Einkaufskorb nicht funktioniert, wechseln die Nutzer zu einem anderen Shop, und die Verkäufe sinken. Wenn die Reaktionszeiten über längere Zeit langsam sind oder häufig Fehler auftreten, werden sogar treue Nutzer irgendwann aufhören, das Produkt oder den Dienst zu nutzen. Jede Organisation weiß, was Ausfallzeiten für ihren Ertrag bedeuten. Die mittlere Erkennungszeit (MTTD) und die mittlere Wiederherstellungsfrist (MTTR) haben ihren Preis.



## Moderne, cloudnative Apps sind komplex

Angesichts der Art, wie Anwendungen entwickelt und bereitgestellt werden, und der aus dieser Umstellung resultierenden Komplexität, ist eine umfassende Leistungsüberwachung noch dringender erforderlich.

Die Einführung von Microservice-Architekturen hat die Art und Weise verändert, wie Anwendungen erstellt und gepflegt werden, da diese Architekturen kontinuierliche Integration und Skalierbarkeit in bisher unvorstellbarem Ausmaß ermöglicht haben. Aber die verteilte und vielsprachige (in vielen unterschiedlichen Sprachen und Frameworks geschriebenen) Natur dieser Architekturen bedeutet auch zusätzliche Komplexität. Für kontinuierliche Integration und Automatisierung sind häufige Codepushes erforderlich, und die Leistungsüberwachung ist wichtiger als je zuvor, um Probleme schnell patchen oder Änderungen rückgängig machen zu können, wenn ein Deployment nicht plangemäß verlaufen ist.

Mit cloudnativen Entwicklungsmethoden können Teams kontinuierlich hervorragende digitale Erlebnisse für ihre Benutzer bereitstellen, schnell auf Feedback reagieren und bei Bedarf umschwenken. Die im Hintergrund arbeitende Maschinerie für die Verarbeitung jeder einzelnen Anforderung bleibt den Nutzern dabei verborgen. Also, wie behalten Teams ihre Anwendungen im Blick?

Wir Menschen können solche Mengen an Daten nicht sinnvoll verarbeiten. Wir brauchen eine organisierte Ansicht mit einem klaren Weg zu den Antworten. Wir brauchen APM.

# Was genau ist APM?

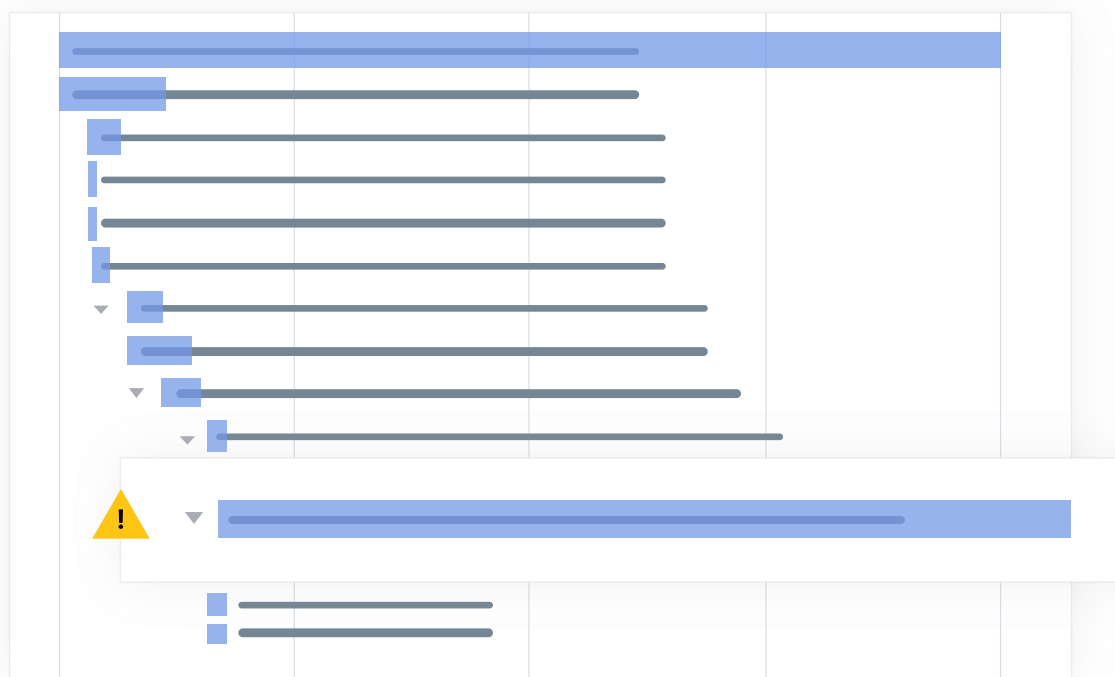
APM ist eine Lösung zum Erfassen, Überwachen und Analysieren der End-to-End-Leistung und des Verhaltens von Transaktionen über Ihre Anwendungen hinweg (und die Dienste, von denen die Anwendungen abhängen). APM misst das Erlebnis der Nutzer, ab dem Moment, in dem sie mit der Anwendung interagieren, bis zum Erreichen der gewünschten Ergebnisse.

Durch diese direkte Verbindung zum Benutzererlebnis ist APM eine Brücke zwischen IT und Geschäftszielen. Die Beziehung ist nicht genau 1:1, hat jedoch eine positive Korrelation. Durch die kontinuierlichen Einblicke in die Leistung Ihrer Anwendungen können Sie mit APM proaktiv arbeiten, anstatt auf Probleme zu reagieren, nachdem Sie über Ihr Ticketingsystem oder sogar über soziale Medien mit Kundenbeschwerden überschwemmt wurden.

## End-to-End-Transparenz

Sobald eine Anwendung instrumentiert wurde, können Sie an den resultierenden Daten exakt erkennen, was in der Anwendung vor sich geht. APM überwacht Transaktionen über deren kompletten Verlauf hinweg und misst sämtliche Anfragen und Antworten, egal wie komplex Ihre Architektur ist.

Anschließend verknüpft APM all diese Traces miteinander und liefert eine komplette Übersicht über die Leistung der Anwendung, von einer sehr allgemeinen Ansicht der Dienste (und deren Interaktionen miteinander) bis hin zu Einblicken auf Codeebene.



*Moment mal ... Haben wir nicht schon Logs, die uns verraten, was in unseren Anwendungen vor sich geht? Logs liefern wertvolle Daten mit kontextbezogenen Informationen über die Art der aufgetretenen Fehler oder Ereignisse. Allerdings zeigen uns Logs nicht die End-to-End-Abläufe des Benutzererlebnisses. Diese Lücken müssen wir mit APM-Traces (und verteiltem Tracing) schließen. Mit der Kombination aus Logs und Metriken erhalten Sie einheitliche Einblicke in Ihr Ökosystem.*

## Latenz und Fehler sichtbar machen

APM überwacht zwei Attribute im Hinblick auf das Benutzererlebnis: Transaktionsdauer und Fehler. Wie lange dauern HTTP-Anforderungen normalerweise? Wie viele HTTP 500-Fehler treten normalerweise pro Stunde auf?

Latenzprobleme treten auf, wenn die Verarbeitung einer Anforderung länger dauert als gewöhnlich. Zum Beispiel wenn ein Nutzer auf ein Produkt klickt, um es sich genauer anzusehen, und das Laden der Antwort mehr als einige Sekunden dauert.

Fehler bedeuten, dass ein unerwartetes Ergebnis aufgetreten ist. Dies kommt vor, wenn eine Anfrage nicht erfolgreich abgeschlossen wird.

Untersuchungen haben immer das Ziel, Probleme zu beheben. Für effektive Untersuchungen müssen Teams in der Lage sein, Ursachenanalysen durchzuführen, um genau herauszufinden, welche Komponente das Problem verursacht. Können wir schnell herausfinden, in welchem Dienst der Leistungsengpass auftritt? Können wir genau bestimmen, welcher Funktionsaufruf oder welche Methode für das Problem verantwortlich ist? Mit APM lautet die Antwort auf beide Fragen „Ja“.

## APM über die Produktion hinaus

Wir erwähnen APM oft im Kontext des Benutzererlebnisses für Anwendungen in Produktionsumgebungen, aber APM kommt sogar noch früher zum Einsatz und wirkt sich auf verschiedene Teams und Umgebungen aus. APM liefert Entwicklern beispielsweise das benötigte Feedback zum Erstellen von Code, mit dem sie die Schleife Entwickeln-Testen-Bereitstellen beschleunigen können.

Viele Unternehmen setzen APM in unterschiedlichen Phasen ein, um die Effizienz der Erstellungs-, QA- und Bereitstellungsphasen ihrer Apps zu verbessern. Wenn Organisationen APM über viele Teams hinweg einsetzen, können sie ihre Prozesse verbessern und effiziente Workflows erstellen, um mehr Zeit für Innovationen zu gewinnen (und weniger Zeit mit Notfallreparaturen zu verbringen).

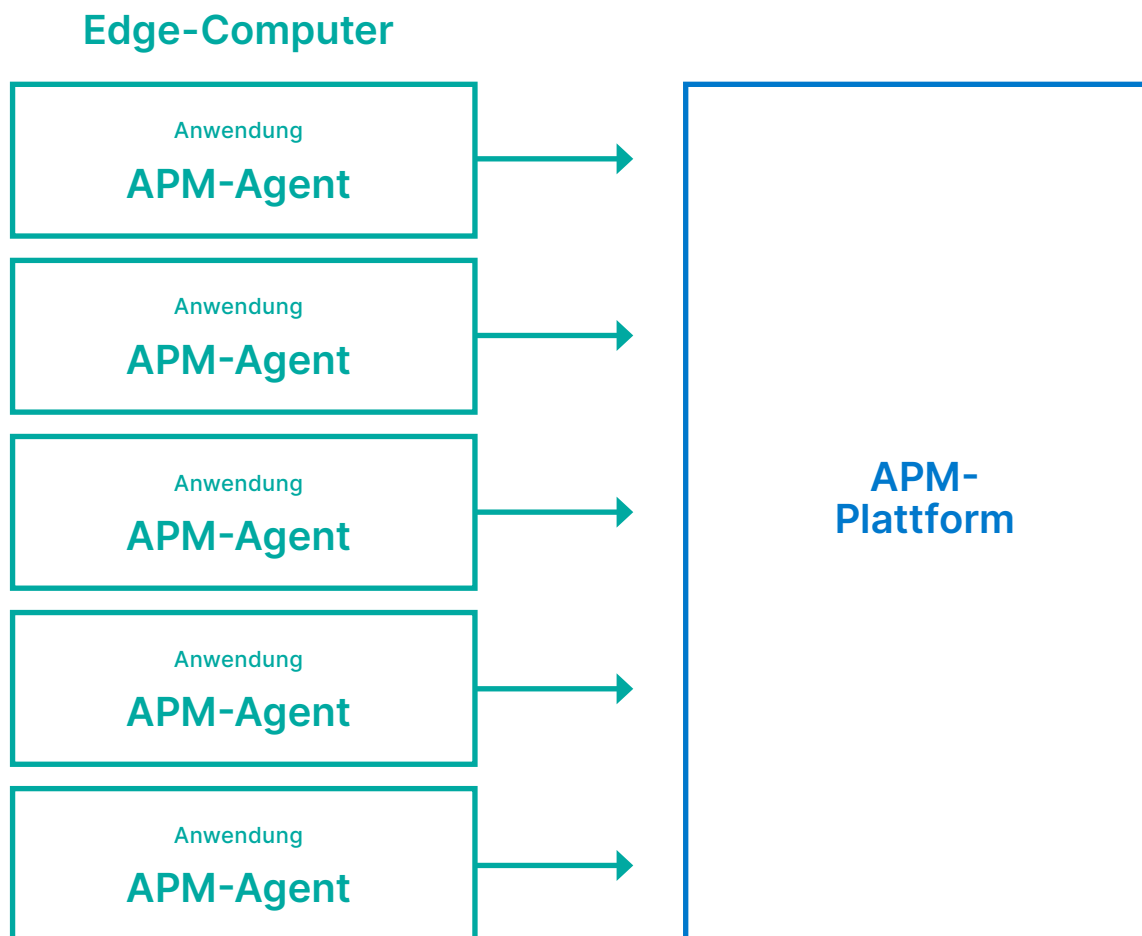
Sie haben also die Vorteile gesehen. Lassen Sie uns jetzt einen Blick auf die Funktionsweise von APM werfen.

# Wie funktioniert APM?

Für APM instrumentieren Sie zunächst sämtliche Aspekte Ihres Systems, die Sie überwachen möchten (Hosts, Container, Anwendungen usw.). Anschließend werden die gesammelten Leistungsdaten zur Analyse an einen Ort Ihrer Wahl übertragen.

## APM-Agents

Ein APM-Agent ist eine Bibliothek, ein Plugin oder eine Erweiterung zur Überwachung der oben beschriebenen Leistungsmetriken. Je nachdem, welche Komponente Sie überwachen möchten (und in welcher Sprache sie geschrieben ist) brauchen Sie möglicherweise einen oder mehrere Agents. Nachdem Sie alle Komponenten identifiziert haben, die Sie überwachen möchten, stellen Sie APM-Agents in den Komponenten bereit. Die Agents hängen zwar vom Anbieter ab, aber die meisten Agents instrumentieren Ihren Code, sammeln Leistungsdaten und senden die Daten dann an einen Server oder einen Collector.



# Instrumentierung und Konfiguration

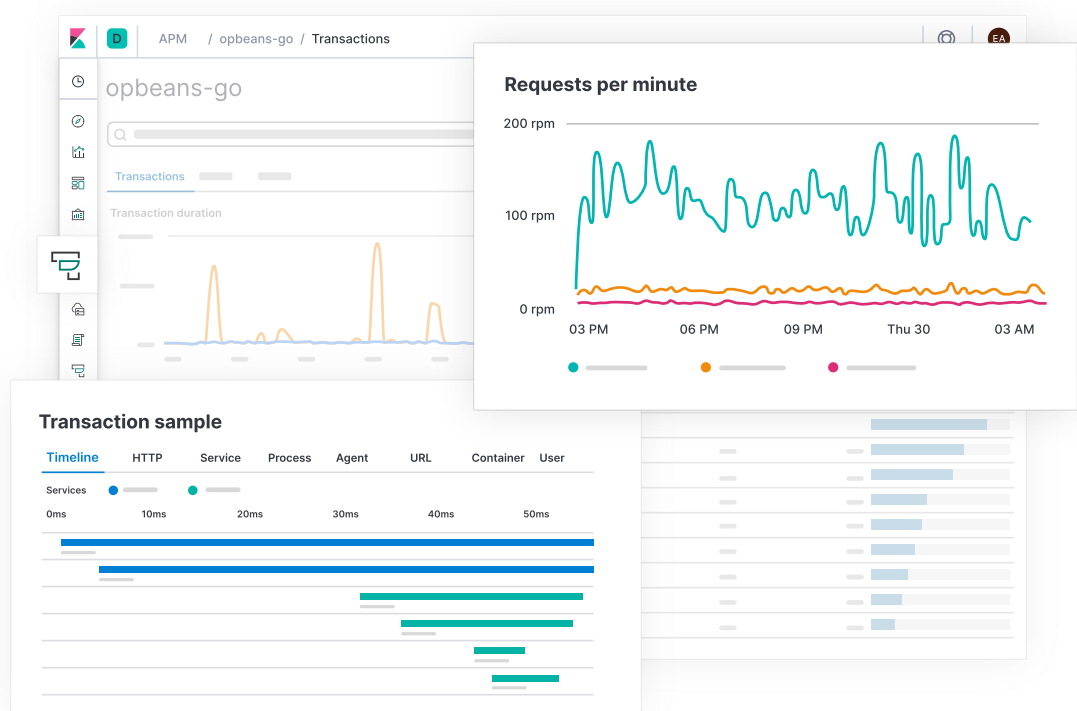
Bei der Instrumentierung erweitern Sie den Code Ihrer Anwendung, um Trace-Daten zu erfassen. Nachdem Sie die erforderlichen Agents installiert haben, können Sie die Teile Ihrer Anwendung instrumentieren, die Sie analysieren möchten, um Leistungsdaten zu erfassen und an einen Ort Ihrer Wahl zu senden, üblicherweise an einen Endpunkt Ihres APM-Tools.

Indem Sie Dinge wie Umgebungsnamen, Samplingraten, Instrumentierungen und Metriken konfigurieren, können Sie es Ihren Teams erleichtern, die im APM-Tool eingehenden Daten zu identifizieren und zu analysieren. Diese Werte konfigurieren Sie normalerweise in der GUI und der API des Tools, oder direkt in den Umgebungsvariablen.

## Analyse

Nachdem die Leistungsdaten an den Ort Ihrer Wahl übertragen wurden, können Sie mit der Analyse beginnen. Die meisten Tools haben eine GUI, in der Sie Fehler, Latenzprobleme und andere Anomalien erkennen können, von denen Ihre Nutzer betroffen sind.

Dies ist ein hervorragender Ort, um gemeldete Probleme zu untersuchen, oder noch besser, um Probleme zu entdecken, bevor sie sich auf Ihre Nutzer auswirken. Identifizieren Sie genau, in welchen Diensten (bis auf die Codeebene) Probleme auftreten, um Ursachenanalysen und MTTR zu beschleunigen.







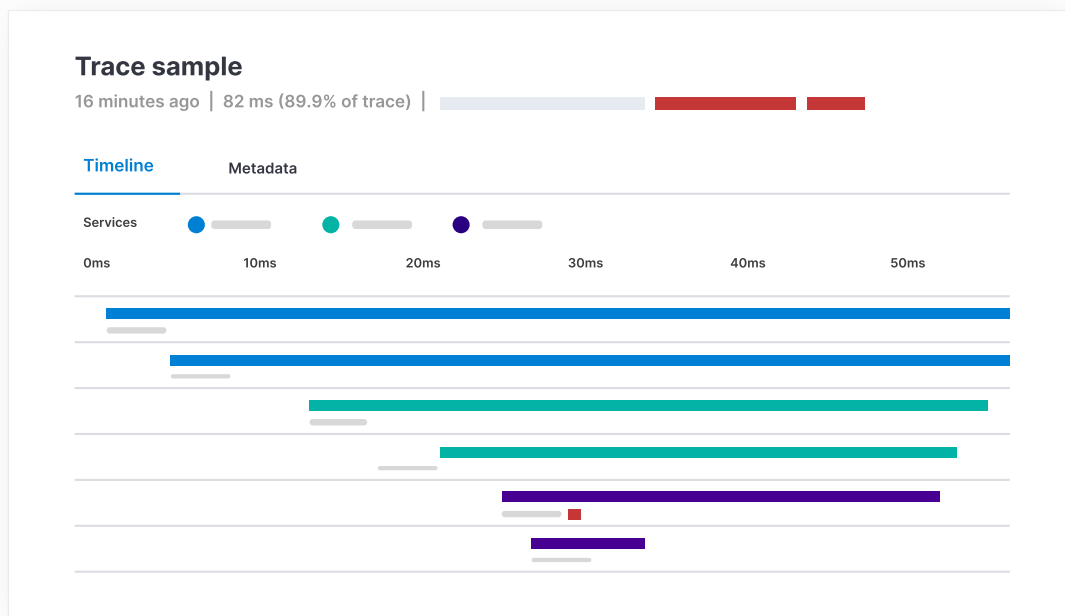
# Wichtige Begriffe

Die einzelnen APM-Tools verwenden möglicherweise etwas unterschiedliche Begriffsdefinitionen, aber im Allgemeinen erfassen sie dieselben Informationen, um Einblicke in die Leistung von Anwendungen zu liefern.

## Verteiltes Tracing

Beim verteilten Tracing werden sämtliche Anfragen nachverfolgt, von der ursprünglichen Webanfrage an Ihren Frontend-Dienst bis hin zu Anfragen an Ihre Backend-Dienste. Auf diese Weise können Sie die Leistung über Ihre gesamte Microservice-Architektur in einer einzigen Ansicht analysieren.

Mit dem verteilten Tracing können Sie mühelos Leistungsengpässe erkennen, indem Sie Ereignisse nach Diensten und anschließend nach einzelnen Anfragen innerhalb des Diensts aufschlüsseln. Sie können verschiedene Fehler und Probleme auf der Codeebene nutzbar identifizieren und Untersuchungen und Ihre MTTR beschleunigen.



# Spans

Alle Einheiten und Komponenten des Workflows werden als Span bezeichnet. Spans sind die Elemente, die Sie normalerweise in der Wasserfallansicht von APM-Analysetools sehen, oft als horizontale Balken. Diese Segmente sind das Herzstück für Ihr verteiltes Tracing. Spans messen die Zeit vom Anfang bis zum Ende einer Aktivität und enthalten Informationen über die Ausführung eines bestimmten Codepfads.

Gängige Attribute von Spans:

- Anfangszeit
- Endzeit
- Name
- Typ

## Transaktionen

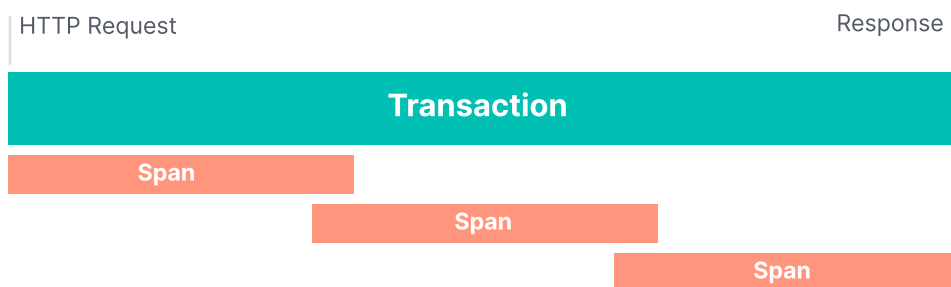
Transaktionen sind eine Art von Span, die ein Ereignis beschreibt und selbst mehrere Spans umfassen kann.

Beispiele für Transaktionen:

- Eine Anfrage an Ihren Server
- Ein Batchauftrag
- Ein Hintergrundauftrag

Transaktionen enthalten zusätzliche Attribute, wie etwa Daten über die Umgebung, in der das Ereignis erfasst wird:

- Dienst: Umgebung, Framework, Sprache usw.
- Host: Architektur, Hostname, IP usw.
- Prozess: Argumente, PID, PPID usw.
- URL: komplett, Domäne, Port, Abfrage usw.
- Benutzer: (falls angegeben) E-Mail, ID, Benutzername usw.

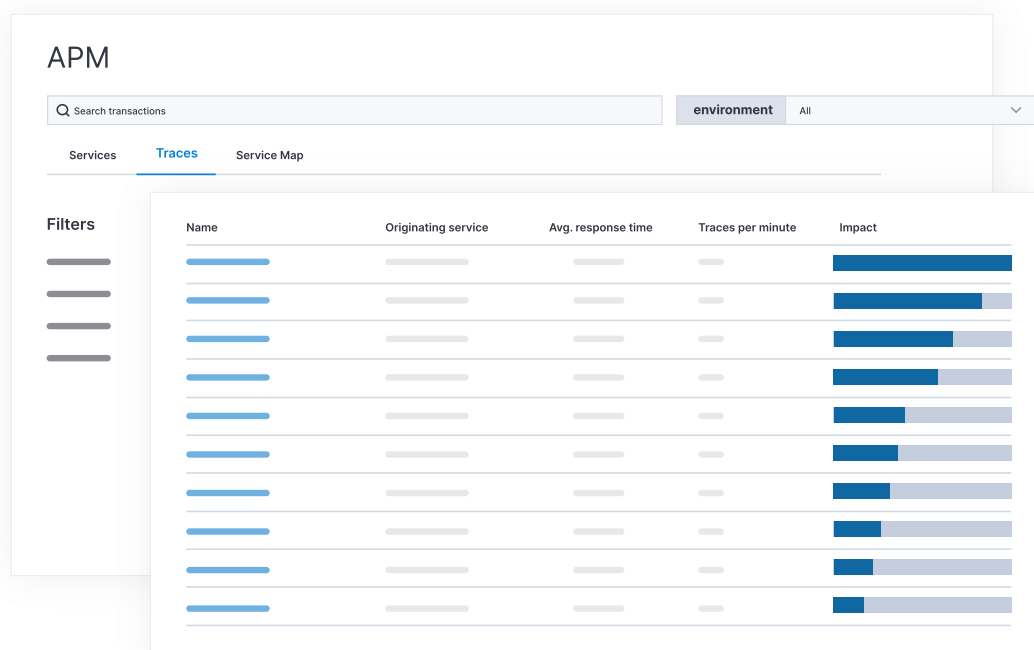


# Traces

Traces messen den Status und die Dauer von Anfragen in Ihrer Anwendung. Spans und Transaktionen bilden gemeinsam einen Trace und schlagen eine Brücke von der Benutzeranfrage bis hin zur Antwort.

Sie werden einzelne Traces für Aktionen wie die Verarbeitung einer Zahlung, die Bearbeitung einer abgeschlossenen Bestellung oder die Aktualisierung des Versandstatus sehen.

In Kombination mit den Logs und Metriken aus der Anwendung (und anderen Aspekten Ihrer Infrastruktur) liefern Traces einen vollständigen Überblick über Ihr gesamtes Ökosystem.



## Real User Monitoring (RUM)

Beim Real User Monitoring (RUM) erfassen wir die direkten Benutzerinteraktionen mit Clients in Produktionsumgebungen. RUM zeigt Ihnen beispielsweise exakt, wie ein Nutzer mit Ihrer Website interagiert hat, zusammen mit den aufgetretenen Antwortzeiten und Fehlern. Dazu wird ein Agent im Browser des Nutzers verwendet, der die Leistungsdaten sammelt und zur Analyse an das APM-Tool überträgt.

# Auswahl eines APM-Tools

Wie bei jeder Toolauswahl sollten Sie zunächst Ihre Anforderungen eindeutig definieren. Berücksichtigen Sie beim Auflisten Ihrer Anforderungen auch Ihr zukünftiges Wachstum und Ihre zukünftigen Bedürfnisse. Planen Sie eine Machbarkeitsstudie (Proof of Concept, PoC), um sicherzustellen, dass das Tool Ihre Anforderungen wirklich erfüllt. Funktionieren die gezeigten und vom Anbieter beworbenen Fähigkeiten in Ihrer Umgebung wie erwartet? Mit einer PoC können Sie Überraschungen und versteckte Kosten in späteren Phasen vermeiden.

## Technische Fähigkeiten

Erstellen Sie eine Checkliste mit den erforderlichen Features für Ihre spezifischen technischen Anforderungen. Erstellen Sie eine gründliche und differenzierte Liste Ihrer Anforderungen. Die folgende Liste ist zwar nicht vollständig, bietet aber einige Denkanstöße:

- ✓ **Unterstützung für Sprachen und Frameworks**
  - Unterstützt das APM-Tool die wichtigsten Sprachen und Frameworks in Ihrer Umgebung?
- ✓ **Tiefe und Umfang der Einblicke**
  - Liefert das Tool Leistungseinblicke auf der Codeebene?
  - Erhalten Sie End-to-End-Einblicke in Ihren Tech-Stack?
- ✓ **Funktionen für Suche, Analytics und grafische Erkundung**
  - Enthält das Tool eine kuratierte GUI für proaktive und reaktive Untersuchungs-Workflows?
  - Enthält es grafische Tools wie Wasserfall- und Abhängigkeitsdiagramme, um die Leistung verteilter Apps analysieren zu können?
  - Bietet es eine flexible (und schnelle) Abfragesprache für Ad-Hoc-Untersuchungen?
  - Erhalten Sie ein flexibles Visualisierungs-Framework, das über die standardmäßig von Anbietern bereitgestellten Dashboards hinausgeht?
- ✓ **Integration in Ihr gesamtes Ökosystem**
  - Lässt sich das Tool nahtlos in die restlichen Standard-Tools in Ihrer Umgebung integrieren, z. B. PagerDuty für Incident-Reaktionen, ServiceNow für die Fallbearbeitung oder Slack für Benachrichtigungen? Wie steht es mit Ihren CI/CD-Frameworks, um die Leistungsauswirkungen von Codepushes und Deployments automatisch nachverfolgen zu können?

## Nutzerfreundlichkeit

Bei der Auflistung der technischen Feature-Anforderungen sollten Sie die Nutzerfreundlichkeit und Barrierefreiheit der einzelnen Features berücksichtigen. Denken Sie an die verschiedenen Rollen, die mit dem Tool interagieren werden, und berücksichtigen Sie deren Erfahrungen, Erwartungen und Workflows, um die Frage nach der Nutzerfreundlichkeit zu beantworten. Betrachten Sie für Ihre Bewertung nicht nur das Einstiegserlebnis, sondern auch den kontinuierlichen Verwaltungsmehraufwand.

- Wie schnell können Sie eine Anwendung instrumentieren und Einblicke erhalten?
- Ist die GUI intuitiv und nutzerfreundlich?
- Wie aufwändig ist die Bereitstellung der Lösung?
- Wie einfach ist es, die Plattform zu aktualisieren? Wie steht es mit den Agents?
- Können die Agents zentral von einem einzigen aus verwaltet werden?
- Wie aufwändig ist die Skalierung der Lösung (in beide Richtungen)?

## Deployment-Optionen

Es ist wichtig, dass das ausgewählte APM-Tool Ihre Vorlieben im Hinblick auf die Softwarenutzung unterstützt.

- Bevorzugen Sie SaaS-Optionen, um Betriebs- und Verwaltungskosten zu senken?
- Möchten Sie eine Option, die Ihnen dabei hilft, Ihre jährlichen Ausgabenverpflichtungen beim Cloudanbieter Ihrer Wahl (AWS, Azure, Google Cloud usw.) zu reduzieren?
- Brauchen Sie ein selbstverwaltetes Angebot, weil Cloudlösungen aus Kosten- oder Compliancegründen nicht in Frage kommen?
- Verfolgen Sie eine Multi-Cloud- oder Hybrid-Strategie und möchten Ihre APM-Lösung näher bei den Workloads ausführen, um Datenübertragungskosten oder die Latenz zu reduzieren?

## Unterstützung für offene Standards und offene Daten

Im Bereich Observability werden zunehmend offene Standards entwickelt und eingesetzt, wie etwa OpenTelemetry (eine Kombination der beiden offenen Standards OpenTracing und OpenCensus) als standardmäßiges, anbieterneutrales Instrumentierungsframework. Diese Initiativen helfen Entwicklern dabei, ihre Anwendung möglichst einfach in eine andere APM-Lösung zu migrieren, ohne sie neu zu instrumentieren.

Wenn Sie vorhandene Apps mit einem offenen Standard wie etwa Jaeger instrumentiert haben, können Sie die Migration mit einem entsprechenden Tool beschleunigen und vereinfachen. Offene Standards verbessern auch die Zukunftssicherheit Ihrer Investitionen.

# Architektur und Skalierbarkeit

Prüfen Sie, ob das Fundament und die Architektur des ausgewählten APM-Tools robust genug für Ihre aktuelle und zukünftige Umgebung sind. Ihre Fähigkeit, Leistungsprobleme in Anwendungen schnell untersuchen und beheben zu können, hängt von der Leistungsfähigkeit Ihres APM-Tools ab. Führen Sie bei der Bewertung Ihres APM-Tools unbedingt Lasttests durch (oder werten Sie Benchmark-Daten aus, falls Lasttests nicht in Frage kommen), um sicherzustellen, dass die Lösung das erwartete Ingestions- und Abfragevolumen in Ihrer Umgebung verarbeiten kann, ohne unter der Last einzuknicken.

- Basiert das Tool auf einer einfachen Architektur? Oder verwendet es intern einen Flickenteppich von Frameworks, der irgendwann nachgeben wird?
- Unterstützt die Lösung Hochverfügbarkeit?
- Lässt sie sich problemlos skalieren, um Lastspitzen im überwachten Datenvolumen verarbeiten zu können?
- Gibt es für das verarbeitete Volumen (Apps, Metriken, Abfragen usw.) irgendwelche Limits?

# Sicherheit

Die Sicherheitsüberprüfung sollte bei Ihrem Tool-Auswahlverfahren eine wichtige Rolle spielen. Berücksichtigen Sie unbedingt die beiden folgenden Aspekte:

1. Das Sicherheitsengagement des APM-Anbieters im Hinblick auf Entwicklung und Bereitstellung
  - Benötigen die in Ihren Anwendungen bereitgestellten APM-Agents übermäßige Berechtigungen?
  - Wird der Datenverkehr zwischen internen Komponenten angemessen verschlüsselt und geschützt?
  - Verwendet das Tool die erforderlichen Zertifizierungen (insbesondere für SaaS-Lösungen)?
  - Verwendet das Tool Erweiterungen von Drittanbietern? Wie sicher sind diese Erweiterungen?
2. Zugriff auf das APM-Tool sichern und kontrollieren
  - Lässt sich das Tool in das Zugriffsverwaltungssystem Ihres Unternehmens integrieren?
  - Unterstützt das Tool eine rollenbasierte Zugriffssteuerung mit differenziertem Berechtigungsmodell?

## Fähigkeiten, die über APM hinausgehen

Denken Sie beim Bewerten von Tools über die typischen Grenzen von APM hinaus. Der Bedarf nach konsolidierten Tools und einheitlichen Einblicken führt dazu, dass viele Organisationen APM mit Logs und Metriken kombinieren – den beiden anderen „Grundpfeilern der Observability“ – um Untersuchungen zu beschleunigen und ihre MTTR zu verkürzen. Überprüfen Sie, ob das APM-Tool diese einheitliche Vision unterstützt und Logs und Metriken im Kontext der Anwendungs-Trace-Daten bereitstellt. Ist es in der Lage, Daten außerhalb von üblichen IT-Umgebungen zu ingestieren, wie etwa Sentimente aus sozialen Netzwerken oder Supportanfragen von Kunden?

## Preise

Zuletzt sollten Sie auch die Preisooptionen sorgfältig abwägen, um sicherzustellen, dass das Tool Ihrer Wahl Sie nicht zu Kompromissen im Hinblick auf Ihre Transparenz- oder Überwachungsziele zwingt. Wie auch bei vielen der anderen Kriterien sollten Sie sowohl Ihre aktuelle als auch Ihre zukünftige Nutzung (und Architektur) berücksichtigen. Es gibt sehr unterschiedliche Preismodelle für APM-Lösungen: nach der Anzahl von Agenten oder Hosts, nach Hardwareressourcen usw. Manche Anbieter erheben auch zusätzliche Kosten beim Überschreiten bestimmter Schwellenwerte (zum Beispiel für die Anzahl von Containern oder Metriken). Hier sind einige Fragen, die Sie sich im Hinblick auf Preismodelle stellen sollten:

- Entspricht das Preismodell Ihren Geschäftsanforderungen und Ihren Architekturentscheidungen?
- Wie skalieren die Kosten mit Ihrem geplanten Wachstum und Veränderungen an Ihrer Architektur (z. B. von Monolith zu Microservices)?
- Gibt es eine kostenlose Preisebene? Was ist in der kostenlosen Ebene enthalten? Gibt es irgendwelche Nutzungseinschränkungen?
- Welche Supportebene ist im Preis enthalten?



# Über Elastic

Als beliebteste Open Source-Logging-Plattform bringt Ihnen Elastic Geschwindigkeit, Skalierbarkeit und Relevanz für Ihr APM.

Beim Untersuchen von Problemen, die sich auf Nutzer auswirken, zählt jede Sekunde. Mit Elastic APM werden Ihre Leistungsdaten als Index in Elasticsearch gespeichert, damit Ihre Teams Leistungsengpässe in Echtzeit finden können. Elastic APM bietet außerdem Service Maps mit Machine Learning, benutzerdefinierte Warnungsoptionen und mehr, damit Sie Ihren Nutzern ein besseres digitales Erlebnis bieten können. Unter [elastic.co/de/apm](https://elastic.co/de/apm) finden Sie weitere Informationen.

APM ist nur ein Teil des Puzzles. Vereinen Sie Ihre Logs, Metriken und APM-Traces in einer Plattform, um vollständige Observability für Ihr gesamtes Ökosystem zu erhalten. Unter [elastic.co/de/observability](https://elastic.co/de/observability) finden Sie weitere Informationen.



**Kostenlosen Test noch heute starten**