



OpenAPM – make your own solution

Alicia Bondanza, NovaTec Consulting GmbH

Application Performance Management (APM) ist ein wesentlicher Bestandteil der heutigen IT-Landschaft. Bei der Beratungstätigkeit der NovaTec Consulting GmbH trifft man meist auf kommerzielle APM-Suites. Während der Markt für ganzheitliche, sofort einsatzbereite APM-Werkzeuge sicherlich von kommerziellen Anbietern dominiert wird, bietet die Open-Source-Community zunehmend eine Vielzahl von Technologien, die sich alle durch einen individuellen Aspekt von APM auszeichnen. Neben diesen Technologien hat die Community auch APM-bezogene Standards und Frameworks wie OpenTracing oder OpenCensus ins Leben gerufen.

Die Kombination der Aspekte und Stärken verfügbarer Open-Source-APM-Technologien zur Erstellung einer zugeschnittenen APM-Lösung ist je nach ihrem spezifischen APM-Anwendungsfall lohnenswert. Ein solches kombiniertes System kann eine sinnvol-

le Alternative gegenüber kommerziellen Lösungen darstellen oder auch kommerzielle Lösungen in einzelnen Aspekten sehr gut ergänzen. Zum Beispiel bietet Grafana hervorragende Dashboard-Funktionen, die weit über alles hinausgehen, was kommerzielle Lösungen bieten können. Wenn solche Schlüssel-Aspekte für den Anwendungsfall wichtig sind, kann eine benutzerdefinierte APM-Lösung die perfekte Antwort sein.

OpenAPM

Kombinieren und Tailoring erfordern jedoch fundiertes Wissen über die Funktionalität und Interoperabilität von Open-Source-Tools, um sie an die spezifischen Kontexte und Anwendungsfälle anzupassen. Da es viele verschiedene Kombinations- und Integrationsmöglichkeiten der Open-Source-Tools untereinander gibt, ist es eine Herausforderung, geeignete Lösungen zusammenzustellen. Dies führt zu einem hohen Aufwand für die Web-Recherche und zu der Notwendigkeit, verschiedene Prototypen auf eine Art und Weise auszu-testen, die auf „Trial and Error“ beruht. Daher hat die NovaTec GmbH die OpenAPM-Initiative ins Leben gerufen.

OpenAPM fördert die Verwendung offener und maßgeschneiderter APM-Lösungen. Sie konsolidiert und stellt Wissen über die Interoperabilität von relevanten Open-Source-Tools bereit. Darüber

hinaus strebt die OpenAPM-Initiative an, Interoperabilitätslücken zu identifizieren und zu schließen, indem man erforderliche Erweiterungen und Tool-Integrationen sammelt und bereitstellt. Die öffentlich verfügbare, interaktive Landschaft (siehe „<https://openapm.io>“) ermöglicht es, OpenAPM-Lösungen zu skizzieren, indem man die Möglichkeiten der Interoperabilität zwischen relevanten Open-Source-Tools visualisiert.

Um die Einführung einer auf OpenAPM basierenden, angepassten APM-Lösung zu bewerten, sollte man zuerst die verschiedenen Aspekte von APM betrachten und entscheiden, welche für einen gegenwärtigen APM-Anwendungsfall wichtig sind. Es folgt eine Beschreibung der wesentlichen Aspekte, die beim Erstellen von benutzerdefinierten APM-Lösungen berücksichtigt werden müssen.

Aspekte von OpenAPM

Die Idee einer ganzheitlichen APM-Lösung beinhaltet eine Reihe technischer Merkmale und Anforderungen. Auf einer hohen Ebene lässt sich APM als eine Daten-Pipeline betrachten, die in den Aspekten „Datenerfassung“, „Transformation“, „Speicherung“ und „Wertschöpfung“ (Nutzung) aus Performance-Messungen strukturiert ist. OpenAPM ist genau auf diese verschiedenen Aspekte ausgerichtet (siehe Abbildung 1).

Datensammlung

APM verwendet Sonden zur Erfassung von Messungen. Die Instrumentierung passt den Code an, um Laufzeiten zu messen, Methoden-Aufrufe zu korrelieren und Kontextdaten zu extrahieren. Dies geschieht entweder mittels Agenten oder mit Instrumentierungsgerüsten und Bibliotheken:

- Agenten werden der Anwendung beim Start hinzugefügt und als Teil des Anwendungscode ausgeführt, um die gewünschten Daten zu sammeln. In dynamischen Sprachen (etwa Java, .Net) können Agenten die Instrumentierung sogar zur Laufzeit anpassen.

- Instrumentierung, Frameworks und Bibliotheken sind ein jüngeres Phänomen. Sie werden als direkte Abhängigkeiten in den Anwendungscode integriert. Somit sind Messungen Teil des eigentlichen Quellcodes der Anwendung. Die Instrumentierung wird damit zur Kompilierungszeit angewendet.

Die extrahierten Messdaten werden dann zu Kollektoren gesendet, die oft als Hub für einen bestimmten Satz von Agenten dienen. Nach einer möglichen Stapelverarbeitung der Daten leiten die Kollektoren die Daten an die nachfolgenden APM-Komponenten weiter.

Transformation und Speicherung

Bevor man die Daten analysiert, können die erfassten Daten transformiert werden. Transformationstools empfangen Daten in einem Format, reichern diese an und konvertieren sie potenziell dann in eine andere Darstellung. Beispiele sind die Aggregation von Rohdaten oder die Generierung von Zusatz-Information. Komponenten, die Datentransformationen durchführen, empfangen normalerweise Datenverkehr aus mehreren Quellen und können diese auch an mehrere Zielsysteme senden.

Um die Leistung zu überwachen und sie im Laufe der Zeit zu analysieren, müssen die Leistungsdaten dann auf Festplatte oder Arbeitsspeicher gespeichert werden. In APM findet man normalerweise verschiedene Arten von Langzeitspeicher-Systemen:

- Zeitreihen-Datenbanken eignen sich gut zum Speichern großer Mengen metrischer Daten, beispielsweise aggregierter Antwortzeitmaße. Diese Zeitreihen-Daten lassen sich analysieren und transformieren, um beispielsweise Baselines zu berechnen oder Anomalien zu erkennen.
- Bestimmte NoSQL-Technologien eignen sich zum Speichern großer Mengen von unstrukturierten Performance-Daten wie Traces oder Benutzersitzungsdaten. Diese Art von Daten ist wertvoll für die detaillierte Problem-Analyse sowie für die Analyse des Benutzerverhaltens.



Abbildung 1: Integrierte Tools in OpenAPM

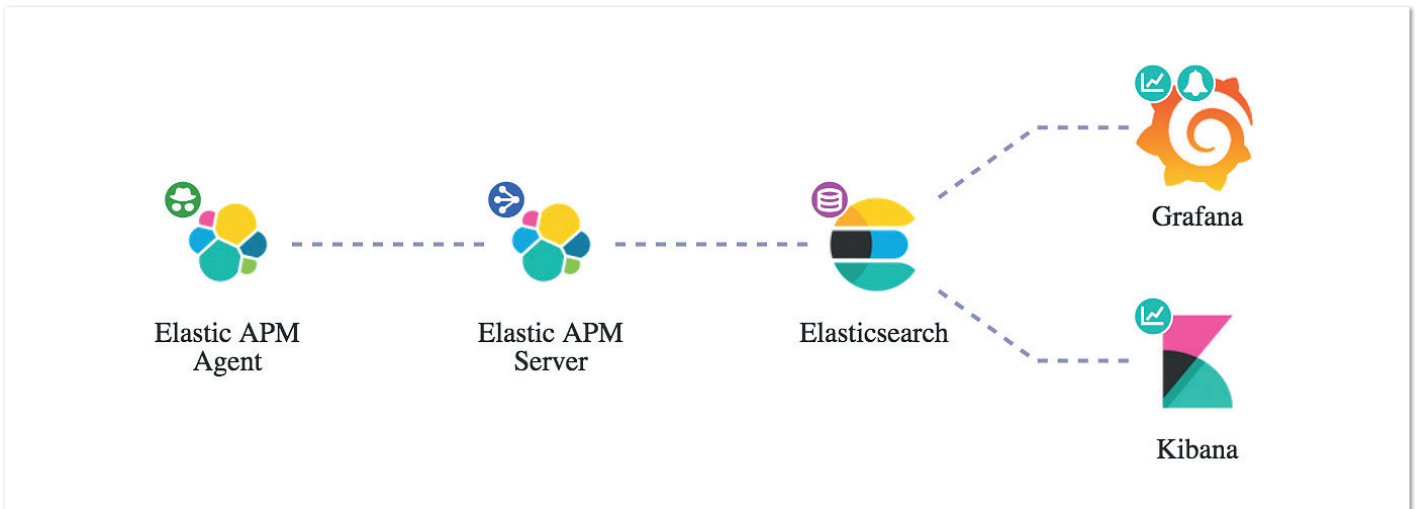


Abbildung 2: Beispiel einer OpenAPM-Lösung

- Relationale Speichersysteme werden auch in APM verwendet, hauptsächlich um Konfigurationsoptionen und Metadaten über die Anwendungslandschaft zu persistieren.

Nutzung und Wertschöpfung

Die Daten stehen jetzt zur Verfügung und zur Nutzung bereit. Um sie zugänglich zu machen, benötigt man zunächst Visualisierungstechnologien, die Benutzer-Schnittstellen bieten, um mit Leistungsdaten, Konfigurationen, APM-Ergebnissen und Erkenntnissen zu interagieren. In APM decken Visualisierungen typischerweise verschiedene Aspekte ab, etwa die Anzeige von Traces, Metriken oder Anwendungs-Architekturen und Flüssen. Mit Dashboards ist es möglich, bestimmte wichtige Leistungs-Indikatoren, die für ein bestimmtes Thema relevant sind, in einer einzigen Sicht zusammenzufassen, etwa eine bestimmte Anwendung oder einen Geschäftsvorgang.

Ein weiterer typischer APM-Anwendungsfall ist die Alarmierung. Alerting ist der Prozess der kontinuierlichen Überprüfung definier-

ter Bedingungen, basierend auf operativen Leistungsmetriken. Ist eine Bedingung nicht erfüllt, senden APM-Systeme entsprechende Benachrichtigungen über Kanäle wie E-Mail, SMS oder Anwendungen von Drittanbietern, um relevante Interessengruppen zu informieren.

Die Open-APM-Landschaft

Nachdem die typischen Aspekte einer APM-Architektur vorgestellt sind, folgt nun ein Beispiel dafür, wie man mit „openapm.io“ benutzerdefinierte APM-Lösungen erstellen kann. Die OpenAPM-Landschaft wird anhand eines Beispiels konkretisiert.

Angenommen, ein Unternehmen verfügt bereits über einen Elastic-Stack. Die Frage stellt sich, welche Open-Source-Tools in den Elastic-Stack integriert werden können, um eine maßgeschneiderte APM-Lösung zu erstellen. Hier kommt Node.js als Programmiersprache für die Anwendungen zum Einsatz. In der OpenAPM-Landschaft besteht die pragmatische Möglichkeit, die Anwendungen zu überwachen, verteilte Traces abzurufen und die

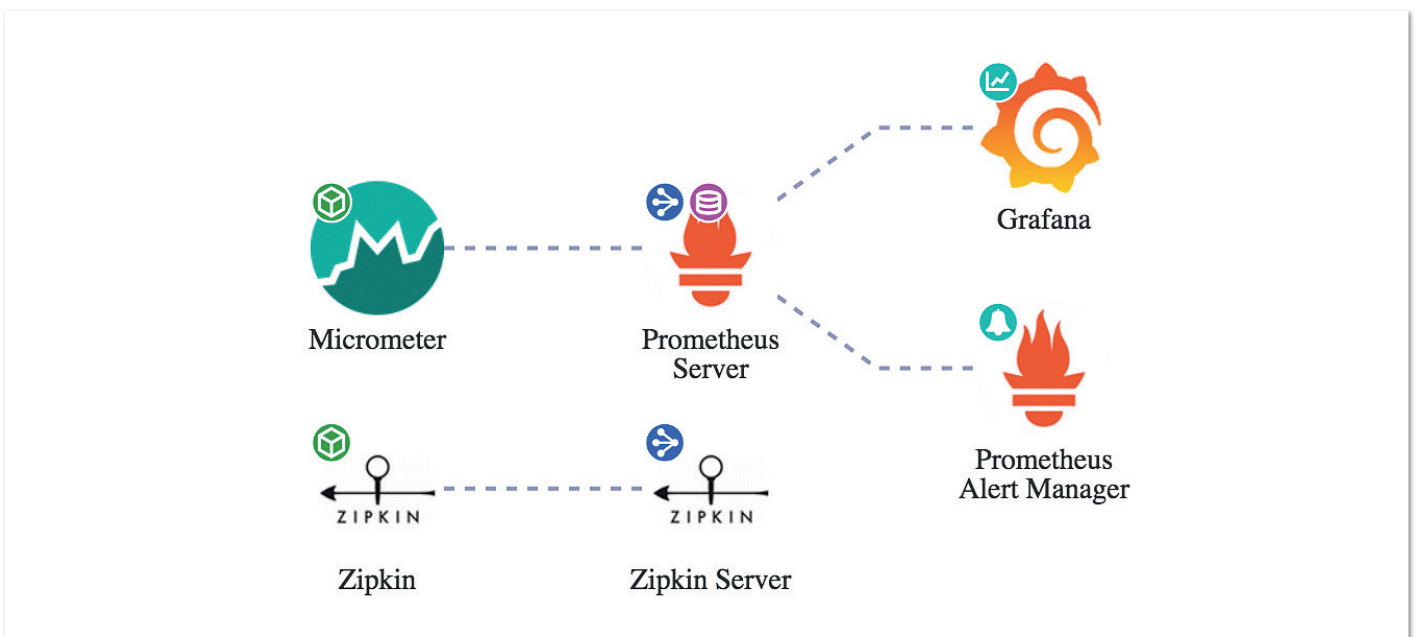


Abbildung 3: Auswahl an Werkzeugen für die APM-Lösung

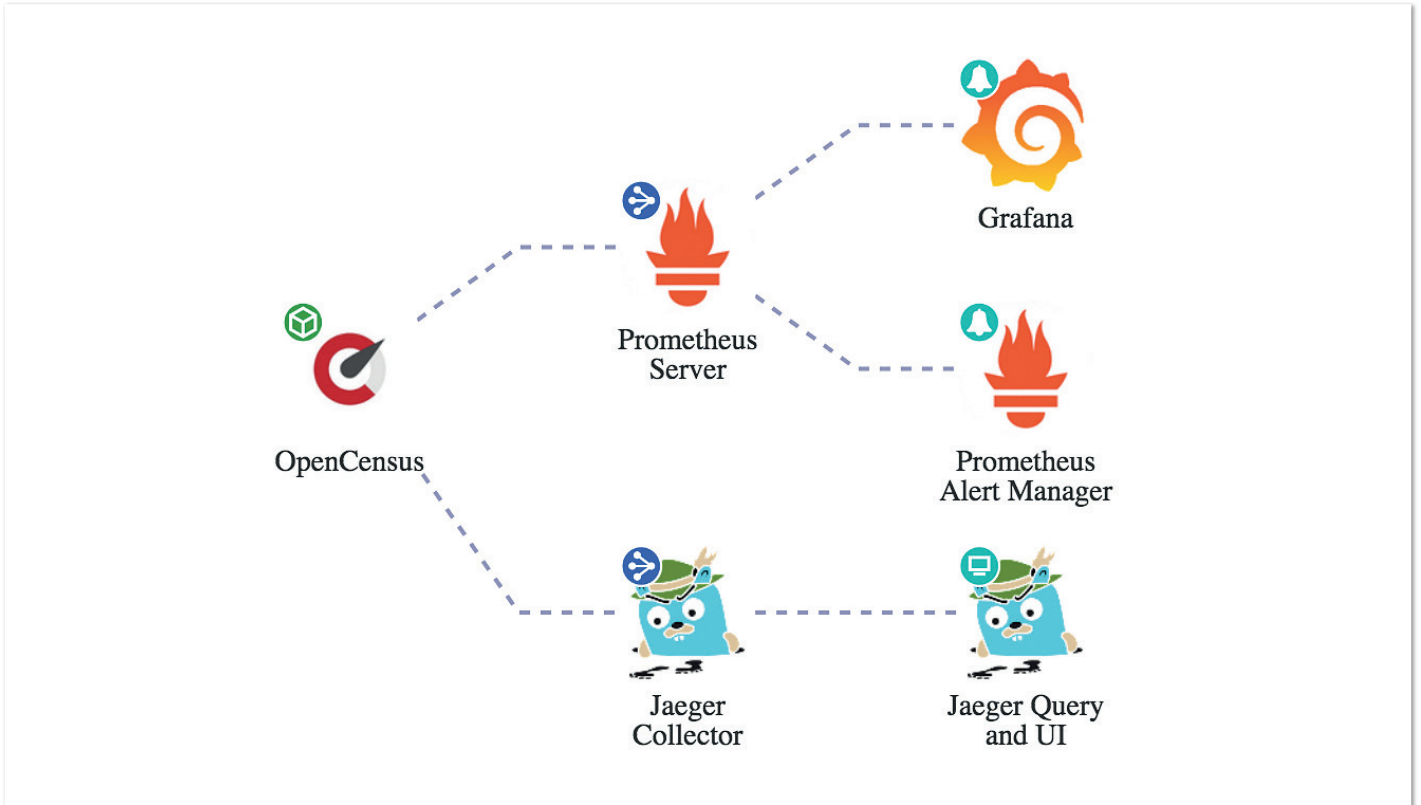


Abbildung 4: Überwachung von Anwendungen

Ergebnisse in Dashboards zu visualisieren. Dazu wählt man zuerst Elasticsearch für die Speicher-Dimension. Danach besteht die Möglichkeit, im Filterbereich nach Node.js und Tracing zu filtern, um nur Tools zu sehen, die in der Lage sind, Tracing-Information aus Node.js-Anwendungen zu erfassen. *Abbildung 2* zeigt eine OpenAPM-Lösung, die zu diesem Beispiel passt. Falls Elastic in Zukunft Java-Monitoring oder Anomaly Detection anbietet, wird die Open-Source-Community die Landschaft entsprechend aktualisieren.

Überwachen von Spring-Boot-Anwendungen

Ein anderes Beispiel ist die Überwachung von Spring-Boot-Anwendungen. Für diese möchte man auch Metriken und Traces überwachen sowie zusätzlich Warnfunktionen integrieren. Nachdem die Landschaft gefiltert wurde, um diese Anforderungen zu berücksichtigen, zeigt sie eine Auswahl von Werkzeugen, aus denen man eine geeignete APM-Lösung zusammenstellen kann (*siehe Abbildung 3*).

Soll stattdessen die Leistung von in Go geschriebenen Anwendungen überwacht werden, führt das Durchsuchen der OpenAPM-Landschaft möglicherweise zu einer Lösung, wie in *Abbildung 4* dargestellt. Die OpenAPM-Landschaft umfasst auch das Open-Source-APM-Tool inspectIT. Möchte man Java-Anwendungen überwachen, könnte ein typisches Setup ebenfalls so aussehen.

Fazit

Mit diesem reichhaltigen OpenAPM-Toolset kann eine zugeschnittene APM-Lösung erstellt werden, die den individuellen Anforderungen entspricht. Dies erfordert natürlich einen gewissen Anpassungsaufwand, abhängig von den spezifischen APM-Anforderungen. Im Vergleich zu kommerziellen APM-Lösungen erhält man jedoch einzigartige Vorteile:

- Die Lösung besitzen und nicht mehr an die Entscheidungen eines Herstellers gebunden sein. Die Lösung kann flexibel und erweiterbar sein und sich im gewünschten Tempo weiterentwickeln.
- Nur für das bezahlen, was man auch wirklich nutzt. Bei kommerziellen APM-Suites stellt man häufig fest, dass Kunden hohe Lizenzgebühren zahlen und viel Zeit in eine Erstkonfiguration investieren. In der Praxis wird jedoch immer nur ein Bruchteil der verfügbaren Funktionen genutzt.
- Wiederverwendung und Integration von Technologien, die in dem Unternehmen bereits vorhanden sind. Man fügt dem vorhandenen Elastic-Stack eines Teams beispielsweise einen Elastic-Agenten hinzu, um mit dem benutzerdefinierten APM zu starten.

Weitere Informationen unter „<https://openapm.io/>“.



Alicia Bondanza

alicia.bondanza@novatec-gmbh.de

Alicia Bondanza ist Digital Content Creator im Bereich Application Performance Management der NovaTec Consulting GmbH. Alicia studiert Sprach- und Textwissenschaften an der Universität Passau und durch ihre Beiträge, bereitet sie einem breiten Publikum das Fachwissen der IT-Berater der NovaTec Consulting GmbH auf.