Das Reaktive Manifest

Veröffentlicht am 16. Septempter 2014 (v2.0)

In unterschiedlichen Zweigen der Softwareindustrie stoßen Organisationen und Entwickler auf wiederkehrende Muster in den Anforderungen und der Implementierung moderner Systeme. Um den heutigen Ansprüchen zu genügen, müssen diese robuster und anpassungsfähiger sein, als es früher der Fall war. Dieser Wandel ist bedingt durch das rapide wachsende technische und soziale Ausmaß der Verwendung von Computersystemen in unserer Gesellschaft.

Bestanden große Anwendungen vor wenigen Jahren noch aus Dutzenden von Servern, die Antwortzeiten im Sekundenbereich lieferten, regelmäßig für Stunden gewartet wurden und Daten in der Größenordnung von Gigabytes verarbeiteten, so umfassen sie heute Tausende von Vielkernprozessoren, verteilt auf mobile Endgeräte und in der Cloud. Benutzer erwarten Antwortzeiten im Bereich von Millisekunden und ständige Verfügbarkeit, die Datenmenge bemisst sich in Petabytes.

Wir glauben, dass die Anforderungen, die heute an Computersysteme gestellt werden, nur zu erfüllen sind durch die gleichzeitige Ausrichtung an vier Qualitäten, deren Wert bislang nur einzeln betrachtet wurde: Systeme müssen stets antwortbereit, widerstandsfähig, elastisch und nachrichtenorientiert sein. Dann nennen wir sie reaktive Systeme.

Computersysteme, die nach diesen Anforderungen entwickelt werden, erweisen sich als anpassungsfähiger, mit weniger starr gekoppelten Komponenten und in jeder Hinsicht skalierbarer. Sie sind einfacher weiterzuentwickeln und zu verändern. Sie reagieren zuverlässiger und eleganter auf Fehler und vermeiden so desaströse Ausfälle. Reaktive Systeme bereiten dem Benutzer durch ihre fortwährende Antwortfreudigkeit eine interaktive und höchst befriedigende Erfahrung.

Reaktive Systeme sind:

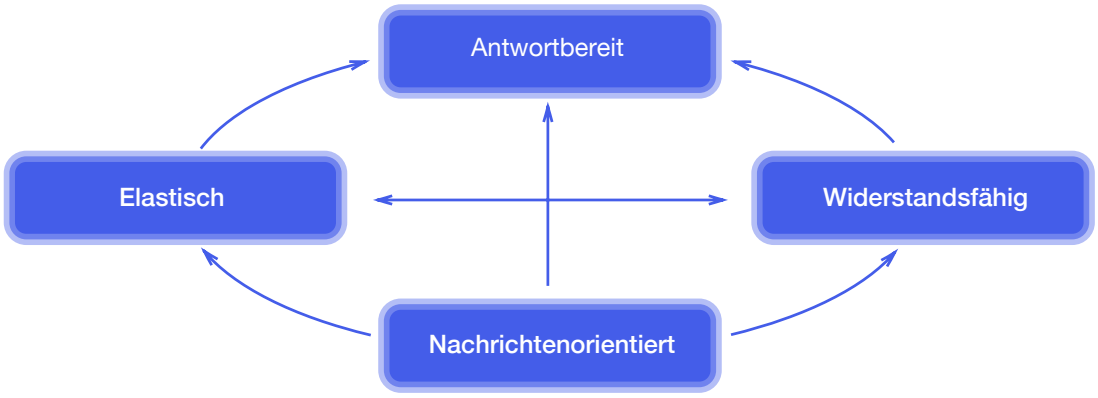
Antwortbereit (Englisch: responsive): Das System antwortet unter allen Umständen zeitgerecht, solange dies überhaupt möglich ist. Antwortbereitschaft ist die Grundlage für Funktion und Benutzbarkeit eines Systems, aber noch wichtiger ist, dass Fehler in verteilten Systemen nur durch die Abwesenheit einer Antwort sicher festgestellt werden können. Ohne vereinbarte Antwortzeitgrenzen ist die Erkennung und Behandlung von Fehlern nicht möglich. Eine weitere Facette ist, dass konsistente Antwortzeiten als Zeichen von Qualität Vertrauen stiften und so weitere Interaktion fördern.

Widerstandsfähig (English: resilient): Das System bleibt selbst bei Ausfällen von

Hard- oder Software antwortbereit. Dies bezieht sich nicht nur auf hochkritische Anwendungen: jedes System, welches nicht widerstandsfähig ist, verliert durch einen Ausfall seine Antwortbereitschaft und damit seine Funktion. Widerstandsfähigkeit ist nur erreichbar durch Replizieren der Funktionalität, Eindämmung von Fehlern, Isolation von Komponenten sowie Delegieren von Verantwortung. So bleibt der Ausfall eines Teilsystems auf dieses begrenzt, andere Teilsysteme sind geschützt und in ihrer Funktion nicht behindert. Die Wiederherstellung des Normalzustandes wird einer übergeordneten Komponente übertragen, die durch die gesteuerte Replizierung der ihr untergeordneten Komponenten die geforderte Verfügbarkeit sicherstellt. All dies bedeutet, dass Nutzer eines auf diese Weise widerstandsfähigen Systems von der Last befreit sind, sich mit dessen Ausfällen auseinandersetzen zu müssen.

Elastisch (Englisch: elastic): Das System bleibt auch unter sich ändernden Lastbedingungen antwortbereit. Auch hier bildet die Verteilung und Replizierung von Funktionalität die Grundlage, auf der das System auf Veränderungen reagiert. Bei Verminderung oder Erhöhung der Last werden automatisch die Replizierungsfaktoren und damit die genutzten Ressourcen angepasst. Um dies zu ermöglichen, darf das System keine Engpässe aufweisen, die den Gesamtdurchsatz vor Erreichen der geplanten Maximalauslegung einschränken. Ideal ist eine Architektur, die keine fixen Engpässe aufweist. In diesem Fall kann das bearbeitete Aufgabengebiet in unabhängige Teile zerlegt und auf beliebig viele Ressourcen verteilt werden. Reaktive Systeme unterstützen die Erfassung ihrer Auslastung zur Laufzeit, um automatisch regelnd eingreifen zu können. Dank ihrer Elastizität können sie auf Speziallösungen verzichten und mit handelsüblichen Komponenten implementiert werden.

Nachrichtenorientiert (Englisch: message driven): Das System verwendet asynchrone Nachrichtenübermittlung zwischen seinen Komponenten zur Sicherstellung von deren Entkopplung und Isolation sowie zwecks Übermittlung von Fehlern an übergeordnete Komponenten. Die explizite Verwendungen von Nachrichtenübermittlung führt zu einer ortsunabhängigen Formulierung des Programms und erlaubt die transparente Skalierung von Komponenten. Die Überwachung von Nachrichtenpuffern ermöglicht kontinuierlichen Einblick in das Laufzeitverhalten des Systems — sowohl zur Diagnose als auch zur automatischen Ressourcensteuerung — sowie Priorisierung und Kontrolle der Nachrichtenflüsse. Ortsunabhängigkeit bedeutet, dass Code und Semantik des Programms nicht davon abhängen, ob dessen Teile auf demselben Computer oder verteilt über ein Netzwerk ausgeführt werden. Nicht-blockierende nachrichtenorientierte Systeme erlauben eine effiziente Verwendung von Ressourcen, da Komponenten beim Ausbleiben von Nachrichten vollständig inaktiv bleiben können.



Große Anwendungen bestehen stets aus mehreren Komponenten und sind daher abhängig von deren Reaktivität. Deshalb müssen die genannten vier reaktiven Qualitäten in der Architektur einer jeden Ebene des Gesamtsystems berücksichtigt werden, wodurch reaktive Systeme schichtweise komponierbar sind. Die größten Computersysteme der Welt basieren bereits auf diesen Prinzipien und dienen Milliarden von Menschen in deren täglichen Leben. Es ist an der Zeit, diesen Ansatz bewusst zu Grunde zu legen anstatt ihn in Teilen für jedes Projekt neu zu entdecken.