

Microservices, ein allgemeines Bild

Der Architekturstil von Microservices ist ein Ansatz, bei dem ein System als eine Reihe unabhängiger und lose gekoppelter Dienste erstellt wird, die mit verschiedenen Programmiersprachen und Datenspeichertechnologien erstellt werden können. Das Konzept der Microservices ermöglicht es, eine schwache Konnektivität von Diensten während der Arbeit am System aufrechtzuerhalten, die durch die Muster von Low Coupling und High Cohesion bestimmt wird.

Monolith vs Microservices

Bei einer monolithischen Architektur besteht das System in der Regel aus 3 Blöcken: der Benutzeroberfläche, dem Data Warehouse und dem Serverteil. Der Serverteil verarbeitet Anfragen, führt Geschäftslogik aus, arbeitet mit der Datenbank, füllt HTML-Seiten. Jede Änderung im System führt zu einer Aktualisierung der Version des Anwendungs-Backends. Bei einer Microservices-Architektur wird nur der geänderte Service aktualisiert. Wenn sich die Änderungen auf die Schnittstelle des Dienstes auswirken, erfordert dies die Koordination aller seiner Clients. Das Ziel einer guten Microservices-Architektur ist es, den Bedarf an Service-Koordination zu minimieren.

Was ist ein Vertrag?

Ein Vertrag ist eine Formalisierung der Möglichkeiten der Interaktion mit einem Microservice. Im Falle einer REST-API sind die Endpoints und das Datenschema ein Vertrag. Das anfängliche Design der Architektur ist die Zerlegung des Systems in lose gekoppelte Dienste, die Schaffung von Schnittstellen und Verbindungen zwischen ihnen, die Aufrechterhaltung der Datenintegrität ohne Leistungsverlust. Tolerant Reader und Consumer-Driven Contracts Mustern können bei dieser Aufgabe helfen.

Microservice-Team

Ein Team sollte nicht mehr Personen umfassen, die zwei Pizzen sättigen können. Diese Regel wurde von Amazon beim Sägen seines Monolithen im Jahr 2002 verwendet. Die Developer-per-Service-Regel, d. h. ein Entwickler pro Microservice, ist ebenfalls akzeptabel.

Wenn ein großes System zusammenbricht, kommt es oft vor, dass Teams auf der Basis von Technologie gebildet werden. In diesem Fall platzieren Befehle Logik auf den Ebenen des Systems, auf die sie Zugriff haben. Conway's Law in Aktion:

"Jede Organisation, die ein System entwirft, erhält ein Design, dessen Struktur die Struktur der Teams in dieser Organisation kopiert."

Der Microservice-Ansatz beinhaltet die Aufteilung des Systems in Services entsprechend den Geschäftsanforderungen. Die Services umfassen einen vollständigen Satz von Technologien: UI, Storage, Backend. Dies führt zur Bildung von funktionsübergreifenden Teams, die über genügend Kompetenzen verfügen, um alle notwendigen Dienste zu implementieren und 100% der Geschäftsfunktionalität abzudecken.

Wie groß der Microservice sein sollte

Die Logik des Dienstes sollte vollständig in den Kopf eines Entwicklers passen, unabhängig von der Menge an Code und Personen. Beim Entwerfen eines Systems hat man die Wahl, wie jeden Microservice zu entwickeln. Zum Beispiel:

Node.js — für einfache Seiten mit Berichten.

C++ — für Echtzeitanwendungen.

Python — für die Datenanalyse.

Golang — für Hochlastdienste.

Java — für die Integration in das Enterprise.

Die Architektur des Microservices gibt völlige Freiheit bei der Wahl der Technologien und Tools.