

Composed Applications

Fachlichen Softwarekomponenten wiederverwenden

Viele Konzepte in der Softwareentwicklung befassen sich mit der Wiederverwendung von technischen Softwareartefakten. Composed Applications beschreibt das bisher weitreichendste Konzept, um Anwendungen aufzuteilen und die entstehenden fachlichen Softwarekomponenten wiederzuverwenden. Damit lässt sich Fachlichkeit beliebig interpretieren und neu kombinieren.

Definition

Das Konzept Composed Applications beschreibt das Aufteilen von Anwendungen in fachliche, funktionell abgeschlossene Komponenten. So entstehen Composable Components. Diese fachlichen Komponenten werden mittels einer Integrationsschicht - dem Composable Component-Adapter - eingebunden und lassen sich so in verschiedenen Anwendungen wiederverwenden.

Im Gegensatz zu Modulen oder Bibliotheken nehmen diese Komponenten eine aktivere Rolle. Sie klinken sich direkt in das Prozessmodell der Hauptanwendung

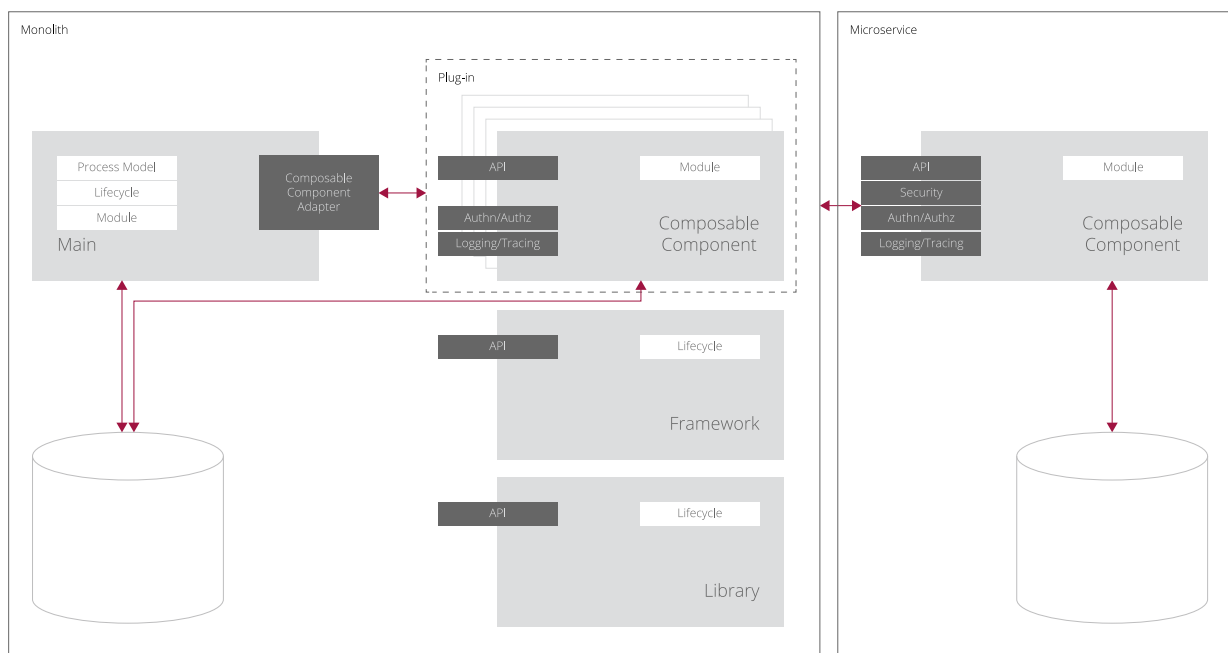
ein und erweitern damit die Fachlichkeit der Anwendung. Die Integrationsschicht bietet verschiedene Schnittstellen, um die Einbindung zu gewährleisten. Neben APIs kommen auch Event-Busses, Callback-Mechanismen und ähnliches zum Einsatz. Durch die funktionelle Abgeschlossenheit müssen auch querschnittliche Themen wie Benutzerauthentifikation und -autorisation sowie Logging und Tracing abgebildet werden.

Das Konzept der Composed Application funktioniert in einem monolithischen Ansatz, in dem die Softwarekomponenten in derselben Laufzeitumgebung aus-

geführt wird, etwa im Hauptprozess. Aber auch in verteilten Ansätzen, wo die Softwarekomponenten in einen Microservice ausgelagert sind. Bei dem verteilten Ansatz muss zusätzlich die Kommunikation zwischen den einzelnen Services abgesichert werden.

Referenzszenario

Ein Unternehmen entwickelt eine Softwarekomponente, deren Fachlichkeit in mehreren Anwendungen benötigt wird, beispielhaft eine Berechnungsfunktion. Mittels einer Integrationsschicht soll die Komponente in verschiedene Anwendungen des Unternehmens integriert wer-



Wiederverwendung

- eine fachliche Softwarekomponenten in verschiedenen Anwendungen
- langfristig sinkender Entwicklungsaufwand
- stabilere Softwarekomponenten

Flexibilität

- einfacher Austausch
- zentrale Steuerung
- parallele Entwicklungen



Lebenszyklus

- unterschiedliche Alterungsprozesse
- alternative Implementierungen
- vereinfachte Fehlerkorrektur

Entwicklungsorganisation

- unabhängige Teams
- klare Verantwortlichkeiten
- zentrale Ansprechpartner

den. Das Unternehmen kann sich jetzt auf die Entwicklung einer Implementierung konzentrieren, anstatt für jede Anwendung eine eigene Implementierung der Berechnungsfunktion zu pflegen. Durch funktionale Abgeschlossenheit ist es möglich, fachliche Änderungen und Fehlerbehebungen zeitnah an der Softwarekomponente umzusetzen und dann mit allen Anwendungen auf einmal zu verteilen. Auch alternative Implementierungen, etwa wegen grundlegender Änderungen der Berechnungen, lassen sich so bereitstellen und einbinden.

Implementierungen und die funktionale Abgeschlossenheit lassen sich die Softwarekomponenten schneller aktualisieren sowie korrigieren und sind dadurch stabiler als im Vergleich zu vielen Einzelimplementierungen. Zusätzlich ermöglicht das Konzept eine Parallelentwicklung von Softwarekomponenten.

Potenzial

Besonders bei Unternehmen mit einer großen Anwendungslandschaft lassen sich bei konsequenter Umsetzung des Konzeptes der Composed Applications Kosteneinsparungen erzielen. Durch die Verwendung derselben Softwarekomponente anstatt vieler unterschiedlicher

Reifegrad

Auf der technologischen Seite haben Composed Applications eine breite und ausgereifte Basis. So sind Komponenten-Frameworks, Event-getriebene Architekturen und Callback-Mechanismen bewährte Technologien und schon seit Jahren im Einsatz.

Anders sieht es auf der konzeptionellen Ebene aus. Dort steht die Entwicklung erst am Anfang. Stand heute muss jedes Projekt, das Composed Applications umsetzen möchte, einen neuen und

womöglichen individuellen Lösungsansatz entwickeln. Einen Blueprint für verbreitete Anwendungsszenarien gibt es noch nicht.

Marktübersicht

Direkte Produkte für Composed Applications gibt es nicht. Aber alle Technologien mit Komponentenorientierung stützen und ermöglichen das Konzept der Composed Applications. Für den Technologiebereich Java sind hier das CDI- und EJB-Framework für Jarkarta EE sowie das Spring-Framework zu nennen. Hinzu kommen Mikroframeworks wie Quarkus, Micronaut und Helideon.

Alternativen

Eine Alternative zum Konzept von Composed Applications gibt es nicht - lediglich den Verzicht.

Fazit

- + wiederverwendbare Fachlichkeit
- + begünstigt fachliche Tests
- + höhere Flexibilität
- + einfachere Wartung
- + aufgeteilte Komplexität
- + langfristige Kostensenkung
- hoher initialer Konzeptions- und Entwicklungsaufwand
- hoher Integrationsaufwand bei bestehenden Anwendungen
- nur bei Wiederverwendung sinnvoll



Buzzword Factor (Ent./Customer)

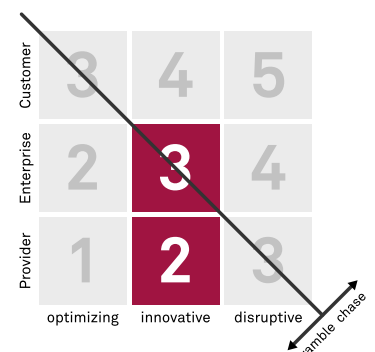
1 low	2 medium	3 high
-------	----------	--------

Entry Barrier (Provider)

1 low	2 medium	3 high
-------	----------	--------

Benefit Level (Provider)

1 low	2 medium	3 high
-------	----------	--------



<https://msg.direct/techrefresh>

Stand: Oktober 2022

msg systems ag