

Cloud-Native Publish/Subscribe Systems with CQRS

Um die Vorteile der Cloud-Plattformen voll auszuschöpfen, sind ereignisgesteuerte Architekturen (EDA) in Kombination mit Event Sourcing und dem CQRS-Pattern (Command Query Responsibility Segregation) eine beliebte Konstellation für die Implementierung. Diese Konstellation nutzt häufig das Publish/Subscribe-Pattern für die interne Kommunikation innerhalb der Applikation, worauf sich auch in hohem Masse verlassen wird.

Die CAOS AG entwickelt eine Identitäts- und Zugriffsmanagementlösung namens ZITADEL, basierend auf der bereits vorgestellten Konstellation. Das Publish/Subscribe-Pattern wurde als individuelle brokerlose Lösung implementiert. Es stellte sich die Frage, ob die Implementierung eines Publish/Subscribe-System mit einem integrierten Broker besser geeignet ist.

Daher werden in dieser Arbeit beide Implementierungsmöglichkeiten mittels Benchmarking verglichen und anschliessend als Vergleichsfaktor genutzt. Wobei beide Möglichkeiten in das Produkt ZITADEL implementiert wurden. Für das Benchmarking wurde die tatsächliche Last der Produktivumgebung analysiert und in einem eigens erstellten Event Generator abgebildet zur Simulation der Last. Während der Lastsimulation wurden Metriken aufgezeichnet und anschliessend als Vergleichsfaktor verwendet.

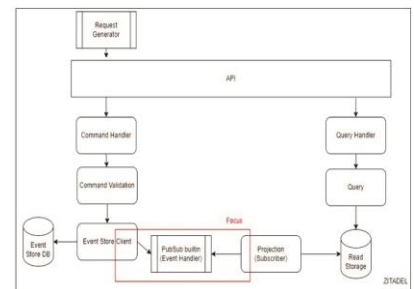
Allgemein betrachtet verbrauchen die Publish/Subscribe-Systeme mit einem integrierten Broker in einer zuverlässigen Umgebung mehr Systemressourcen wie z.B. CPU und Speicher. Ausserdem erhöht sich die Netzwerklast durch den Verkehr und der betriebliche Overhead wird durch die zusätzlichen Komponenten ebenfalls grösser. Hingegen in einer unzuverlässigen Umgebung sind Publish/Subscribe-Systeme mit einem integrierten Broker ressourcenschonender.

Die Entscheidung, welche Art von Möglichkeit implementiert werden soll, hängt jedoch stark von den Anwendungsanforderungen und den Anforderungen an die Publish/Subscribe-Implementierung ab. Im Allgemeinen wird vorgeschlagen, dass für einfache Anforderungen ein Publish/Subscribe-System mit einem integrierten Broker am besten geeignet ist und für komplexere Anforderungen eine benutzerdefinierte Implementierung ohne Broker die bessere Lösung ist.

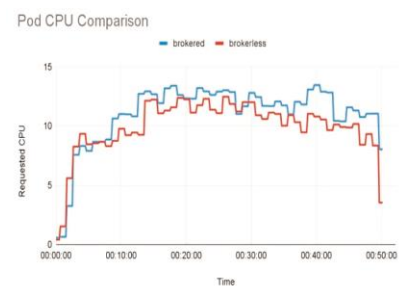


Diplomierende
Silvan Reusser
Darryl Thurnherr

Dozent
Josef Spillner



Übersicht der einzelnen
Komponenten des "System Under
Test"



Gegenüberstellung der CPU
Auslastung während eines Testlaufs