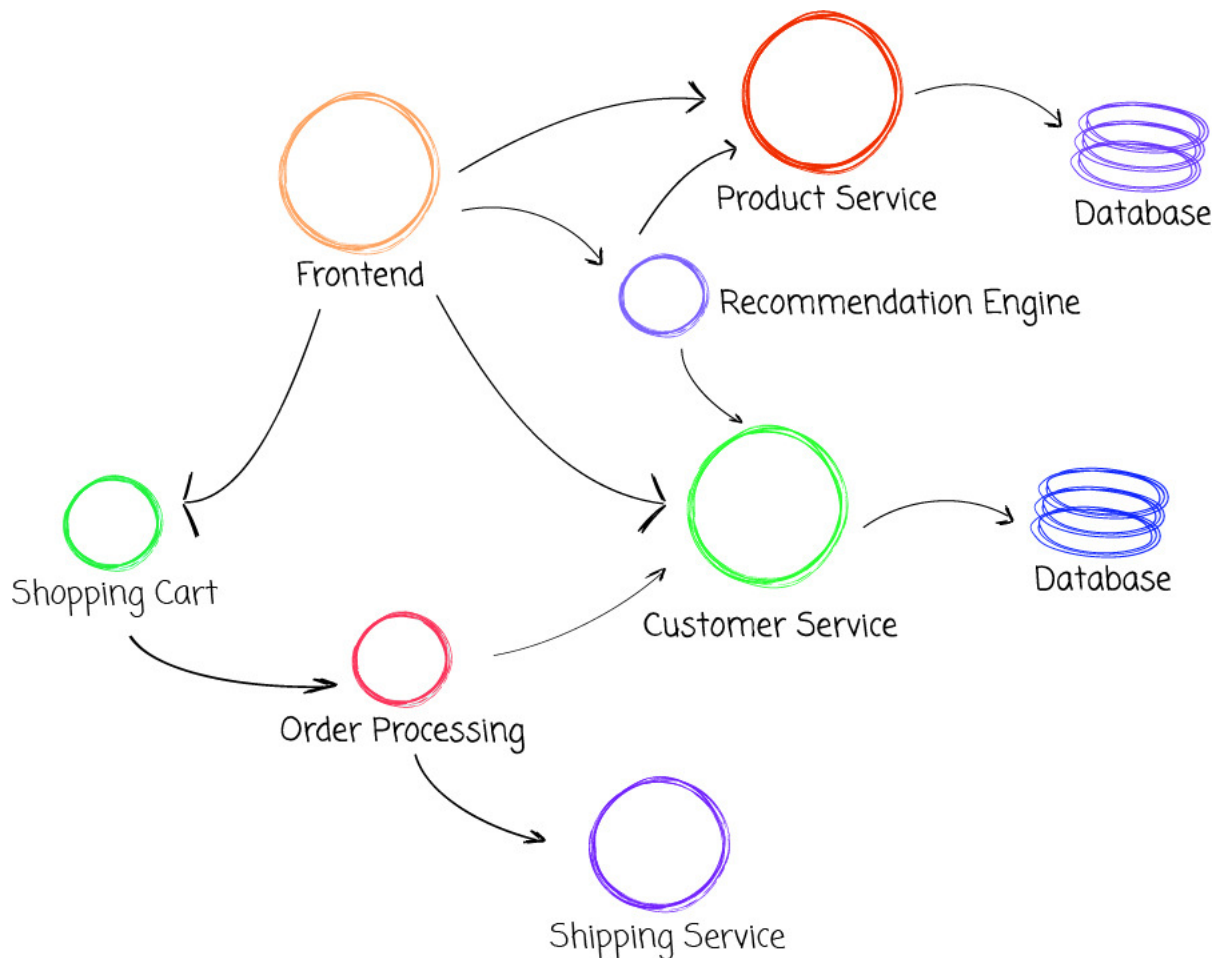


## Microservices, event-driven, reactive: Vert.x



**2015** was het jaar van Microservices. Microservices is een software architectuurstijl waarbij applicaties worden opgebouwd uit onafhankelijk deploybare services. De voordelen hiervan zijn onder andere de mogelijkheid tot het snel ontwikkelen van functionaliteit met een technologie die hier het best voor geschikt is, het onafhankelijk schalen van de verschillende services en het hergebruiken van de services voor diverse doeleinden. Uitdagingen zijn er op het gebied van configuratie, integratie, monitoring en het beheer van de diverse API's.

Vert.x is een toolkit voor het bouwen van Microservice applicaties voor de JVM. Vert.x is toegankelijk voor een grote groep ontwikkelaars, door de ondersteuning van diverse talen, waaronder Java, Groovy, Javascript en Ruby. Vert.x werkt op basis van events die verstuurd worden via de event-bus. De event-bus kan desgewenst over meerdere servers in een gedistribueerde omgeving worden gebruikt. De event-bus maakt gebruik van JSON en werkt via TCP/IP, wat het mogelijk maakt om ook niet-vert.x applicaties aan te sluiten. Via websockets of SockJS kan de event-bus ook gebruikt worden om met browser-clients te communiceren.

Vert.x is reactive en non-blocking. Connecties en events worden asynchroon afgehandeld binnen één of meerdere event-loops (doorgaans 1 per CPU core). Dit in tegenstelling tot klassieke applicatieservers, waarbij er per connectie een thread wordt gebruikt. Dit maakt het mogelijk om op te schalen tot veel meer connecties dan bij het thread per connectie model. De event-loop

controleert voortdurend op het binnenkomen van nieuwe events die afgehandeld moeten worden. Voor alle services, in Vert.x termen heten dit verticles, die geabonneerd zijn op dit event wordt de eventhandler aangeroepen. Het is hierbij van belang dat alle operaties, ook (database) I/O op een non-blocking manier wordt afgehandeld, zodat dat de eventloop events kan blijven afhandelen en niet geblokkeerd blijft voor één specifieke afhandeling. Omdat de verticles onderling communiceren via de eventbus, wordt het op deze manier eenvoudig om gedistribueerde systemen te bouwen.

Het concept van het opdelen van applicatiefunctieiteit in kleine, zelfstandige services (Verticles) en de integratie van deze services via een eventbus maakt dat Vert.x perfect aansluit bij de Microservices architectuur en oplossingen biedt voor de uitdagingen die hiermee gepaard gaan. Vert.x past goed in een DEVOPS omgeving omdat het componenten biedt voor metrics, shell toegang tot de applicatie en Docker ondersteuning. Vert.x is toegankelijk voor een grote groep ontwikkelaars door de ondersteuning van diverse talen en is standalone of geïntegreerd met andere frameworks te gebruiken. Het is daarmee de ideale keuze voor het ontwikkelen van event-driven, reactive microservice applicaties op de JVM!