

De twee geheimen van een unified commerce-strategie

05-05-2022 14:00



De retailsector staat vandaag voor een aantal grote uitdagingen. De consument heeft steeds hogere verwachtingen van het serviceniveau van retailers. Zo verwacht de consument een naadloze overgang, wanneer deze bijvoorbeeld een bestelling plaatst met zijn smartphone en die verder wil opvolgen in de webshop op zijn pc of tablet. Daarnaast verwacht de consument ook dat het winkelpersoneel op de hoogte is en kan assisteren met de online bestelling en op maat gemaakte aanbiedingen, promoties en aanbevelingen te krijgen in plaats van generieke reclame.

Daarnaast is het belangrijker dan ooit dat interne applicaties in real-time gegevens met elkaar kunnen uitwisselen, zodat elke applicatie te allen tijde een accuraat beeld heeft van de huidige applicatiedata. Business users nemen geen genoegen meer met applicaties die hen geen *up-to-the-second* actueel zicht geven over de actuele stand van zaken.

Ook moeten retailers meer en meer samenwerken met andere partijen om hun klantbereik te vergroten en zo de omzet te verhogen. Doen ze dat niet, dan gaan concurrenten met hun klanten aan de haal. Denk aan samenwerkingen met grote online retailplatformen zoals bol.com, Zalando of Amazon. Gelukkig beschikken retailers over twee manieren om aan al deze uitdagingen tegemoet te komen, namelijk API's en event-driven architecturen.

API's

De API-revolutie is al een tijdje aan de gang en het is begrijpelijk waarom: API's vergemakkelijken de integratie tussen applicaties over afdelingen heen en maken een samenwerking met externe businesspartners mogelijk. Hierdoor kunnen grotere en nieuwe marktsegmenten bereikt worden. Daarnaast zijn API's essentieel voor het ondersteunen van een omnichannel gebruikservaring voor klanten en bevorderen zij innovatie omdat zij als bouwstenen voor nieuwe innovatieve toepassingen ingezet kunnen worden.

Wat is er allemaal nodig voor een succesvol API-verhaal? Allereerst is het belangrijk om te vertrekken vanuit een weldoordachte API-strategie, gebaseerd op de doelstellingen: hetgeen je als retailer wenst te bereiken met

behulp van API's. De API-strategie zal vervolgens het API-portfolio bepalen: welke data of functionaliteit moet je als API beschikbaar maken om je doelen te bereiken? Dan pas kun je beginnen met het ontwerpen en ontwikkelen van je API's. Dit alles uiteraard volgens de geldende best practices en guidelines.

Tenslotte is een API management platform vandaag een onmisbare component in elk succesvol API-traject. Dit platform laat API's op een gecontroleerde en veilige manier beschikbaar maken en beheren. Het functioneert als de toegangspoort voor API-afnemers en biedt door middel van aanpasbare dashboards uiterst waardevolle inzichten in het gebruik en de prestaties van je API's. Die inzichten zijn onontbeerlijk om te meten in welke mate de vooropgestelde doelen behaald worden en waar er eventueel moet bijgestuurd worden.

Event-driven architectuur

In plaats van een retail-IT-systeem als een organisatie van synchrone API-calls te bekijken, kan het ook gemodelleerd worden als een aaneenschakeling van events. In plaats van applicaties te ontwerpen die andere applicaties vertellen wat ze moeten doen en te wachten op een antwoord, kunnen deze ook zo ontworpen worden dat ze andere applicaties door middel van events laten weten wat er is gebeurd en die applicaties zelf laten beslissen hoe deze moeten reageren. Dit is de essentie van een event-driven architectuur, en resulteert in een flexibeler, meer schaalbaar IT-landschap.

Event-driven architectuur is natuurlijk niet nieuw. Het concept bestaat al tientallen jaren. Hoe komt het dan dat de interesse in deze vorm van architectuur de laatste jaren zo sterk opleeft? In één woord: Kafka. Dit zogenaamde event-streaming platform, oorspronkelijk ontwikkeld door LinkedIn maar nu open-source, biedt een aantal features die ongekende nieuwe scenario's en use cases mogelijk maken. Kafka maakt het mogelijk om events voor onbepaalde tijd op te slaan en event streams binnen het platform te manipuleren.

Door die unieke features kun je met Kafka use cases implementeren die vroeger te moeilijk, te duur of simpelweg onmogelijk waren: real-time inventarisbeheer, moderniseren van legacy IT, ondersteunen van een microservices architectuur, fraudedetectie en nog veel meer. Deze use cases laten retailers hun omzet verhogen door het verbeteren van de klantenervaring, de kosten te verlagen door het verhogen van de operationele efficiëntie en het verminderen van risico's door een betere naleving van de huidige regelgeving.

Niek Jacobsen, integration architect bij AXI