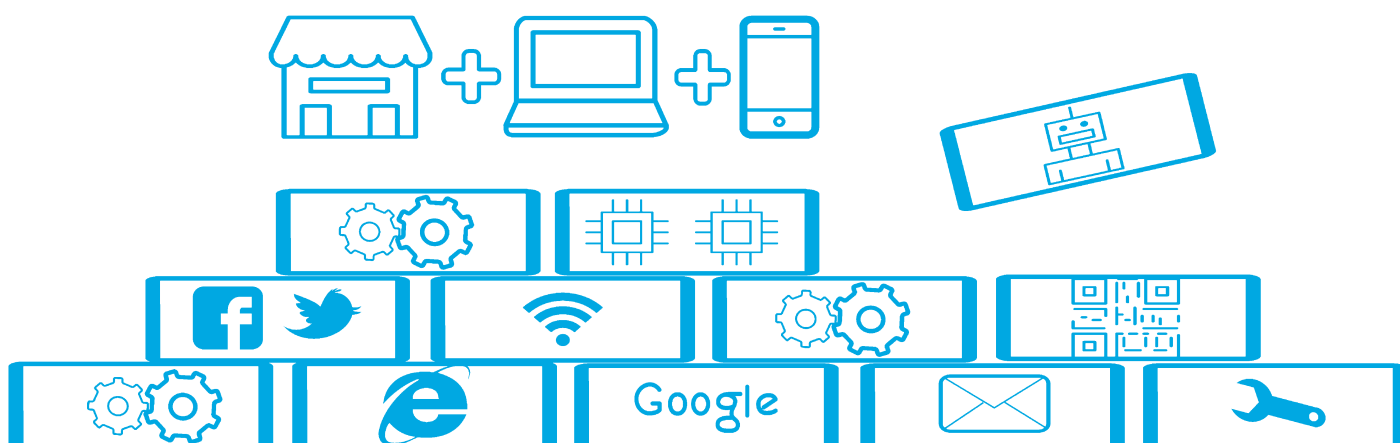


Expertgroep

omnichannel architectuur

Zo moeilijk kan het toch niet zijn?



Gastheer

(v) **hybris**

Voorzitter

Magnus

Zo moeilijk kan het toch niet zijn?

De technologische ontwikkelingen gaan steeds sneller en de eisen aan omnichannel retailers worden steeds groter. Denk aan voorraadinzicht in alle kanalen, relevant gebruik van klantdata, dynamisch omgaan met prijzen en het kunnen integreren met andere verkoopkanalen. Hoe houd je dit met jouw ICT-landschap bij? Dit vraagt om een omnichannel ICT-architectuur. Maar hoe ziet die er uit? Welke ontwerpprincipes liggen daaraan ten grondslag? En hoe integreer je de verschillende systemen en partners in dit landschap? Dit hoofdstuk geeft overzicht, inzicht en aanbevelingen bij de belangrijkste ontwerpvoorbeeldstukken.

1. Onderzoeksvraag en scope

Bedrijven die verkopen aan consumenten opereren de laatste jaren in een steeds veeleisendere omgeving. De moderne 'omnichannel' consument is zeer kritisch en maakt veel gebruik van technologie. Daarnaast ontwikkelt de technologie zich dermate snel dat het bijhouden van ontwikkelingen een uitdaging is. Daarbij is de concurrentie toegenomen, geconsolideerd en geïnternationaliseerd. Dit zorgt ervoor dat de 'papa- en mamawinkel' op de hoek, de bekende oer-Hollandse winkelketens, kleine en grote 'pure players' (van herensokken.nl tot bol.com en Coolblue) en internationale marktplaatsen als Alibaba, Google Shopping en eBay met elkaar concurreren om de gunst van dezelfde consument.

Belangrijke instrumenten in deze concurrentiestrijd zijn tegenwoordig de (online) vindbaarheid, de geboden verkoopkanalen, de actualiteit van content en assortiment, de snelheid van levering, het benutten van data en de geboden klantenservice. Om hierin concurrerend te blijven, is een goede en flexibele omnichannel ICT-architectuur noodzakelijk.

De expertgroep Omnichannel Architectuur stelt zich ten doel bedrijven te ondersteunen bij vragen zoals:

- Hoe ziet de optimale omnichannel architectuur eruit?
- Welke componenten maken er onderdeel van uit?
- Hoe integreer ik die componenten?
- Welke keuzes moet ik daarbij maken, met welke voor- en nadelen?

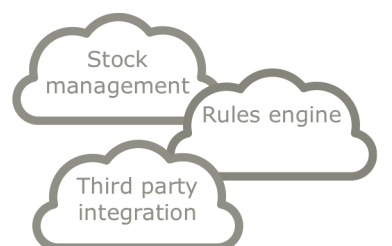
De focus van het onderzoek ligt daarbij op bedrijven die rechtstreeks verkopen aan consumenten en dit doen in een omnichannel omgeving. Met de expertgroep richtten we ons op de relatief grote B2C-bedrijven die fysieke producten leveren, waarbij we de gehele aankoopcyclus (van oriëntatie tot en met aftersales) in beschouwing namen.

2. Doel van de omnichannel architectuur

Het uiteindelijke doel van een omnichannel architectuur is om de klant in alle kanalen optimaal te kunnen bedienen. Om dit meer concreet te maken hebben we verschillende voorbeelden van 'omnichannel customer journeys' gedefinieerd, die de eisen aan een omnichannel architectuur uiteenzetten.

“Kan jouw architectuur het volgende aan?”

- Een consument koopt één verfblik in de winkel en gebruikt zijn loyalty card. Vervolgens koopt hij diezelfde week nog twee extra blikken verf online. Die week geldt de actie “3 voor de prijs van 2”. Bij de online checkout ontvangt deze loyale consument alsnog de actiekorting, omdat zijn offline en online aankopen binnen de actieperiode zijn gecombineerd.
- Een consument heeft zijn klantprofiel bij de retailer gekoppeld aan zijn Facebook-account. Via Facebook plaatst hij een negatieve reactie op een consumentenblog. De klantenservice reageert adequaat naar de consument en voorziet hem van een tegoedbon. Als hij zijn tegoedbon in de winkel inlevert, ziet de winkelmedewerkster in haar systeem de reden van de tegoedbon en informeert bij de consument of alles naar tevredenheid is opgelost. Zij legt vervolgens zijn reactie vast.
- Een consument ziet in de winkel een leuke laptoptas maar de juiste kleur blijkt niet voorradig in dit filiaal. De medewerkster plaatst op haar tablet een bestelling en de consument rekent mobiel af. De order wordt toegewezen aan een ander filiaal waar dit item wel op voorraad is en waarvan op basis van deurteller-gegevens blijkt dat het daar relatief rustig is. Dit filiaal bevestigt de order, print het correcte verzendlabel en verstuurt het item direct naar de betreffende consument.
- Een consument heeft online een gele bikini besteld, maar bij ontvangst thuis blijkt deze toch niet te passen. De dame meldt de retour online aan. Op dat moment wordt de barcode van het retourformulier bij het postbedrijf gekoppeld aan het retouradres: een filiaal in Amsterdam waar deze bikini goed wordt verkocht en de betreffende maat bijna niet meer voorradig is. Het filiaal ontvangt de geretourneerde bikini, stelt vast dat deze nog verkocht mag worden en bevestigt de goede staat van ontvangst in het winkelsysteem. De consument ontvangt vervolgens een e-mail en haar geld terug.





Meer omnichannel journeys zijn uitgewerkt in de volledige blueprint die je kunt downloaden op EcommerceWiki.org

EcommerceWiki/
ShoppingTomorrow

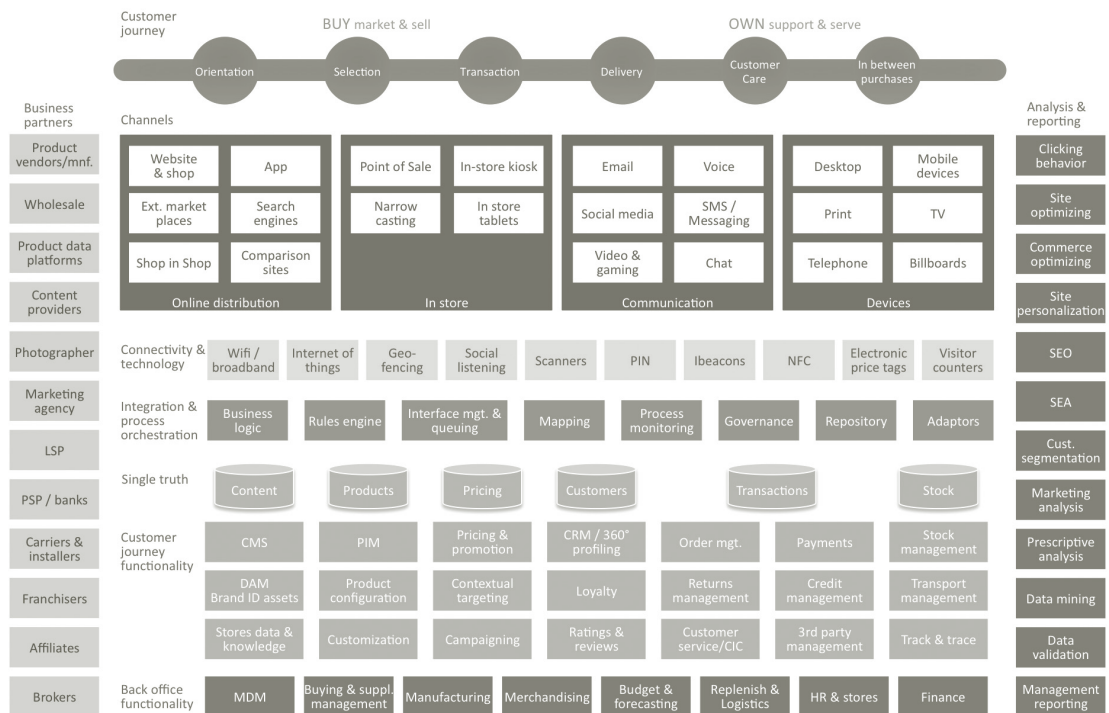
De genoemde (delen van) customer journeys worden nog maar bij weinig retailers volledig door de omnichannel architecturen ondersteund en maken goed duidelijk wat de 'ultieme' eisen aan een omnichannel architectuur kunnen zijn.

Een ideale omnichannel architectuur:

- ondersteunt de gehele customer journey;
- voorziet in alle relevante functionaliteit en informatie;
- maakt alle relevante informatie in alle kanalen beschikbaar;
- heeft voor alle informatie enkelvoudige vastlegging;
- draagt zorg voor een naadloze integratie tussen informatiesystemen en- kanalen;
- is betrouwbaar, beheersbaar, betaalbaar, flexibel en toekomstbestendig.

3. Functionele decompositie

Om bovenstaande en andere 'omnichannel customer journeys' optimaal te ondersteunen zijn zeer veel verschillende functionaliteiten nodig. Onderstaand overzicht brengt de meeste relevante functionaliteiten van een omnichannel retailarchitectuur in kaart.



Overzicht van de meeste relevante functionaliteiten van een omnichannel retailarchitectuur

4. Systeemarchitectuur

Bovenstaande functionele componenten maken onderdeel uit van een of meerdere software-oplossingen. Grote softwarepartijen zijn vaak in staat veel van bovenstaande componenten binnen één oplossing te combineren. Er is echter vrijwel geen enkele organisatie die bovenstaand landschap volledig binnen één pakket heeft ondergebracht. Hiervoor zijn verschillende redenen:

- Er is vrijwel altijd sprake van bestaande systemen ('legacy') die niet gelijktijdig vervangen mogen, kunnen of hoeven worden
- Retailers hebben op deelvlakken dermate specifieke eisen dat hiervoor (kleine maar) specialistische oplossingen nodig zijn
- Grote software-oplossingen kunnen de nieuwste technologische ontwikkelingen op deelgebieden niet meteen 100% geïntegreerd aanbieden
- Bedrijven kiezen bewust voor een architectuur met verschillende software-aanbieders om hun afhankelijkheid en/of investeringen te spreiden.

Hiermee ontstaat in vrijwel elke retailomgeving een 'best-of-breed' systeem-landschap, waarbij het aantal software-oplossingen daarbinnen afhankelijk is van bovenstaande punten.

every.day.counts

In 2014 lanceerde een Europees modehuis een (omnichannel) proeftuin. Het nieuwe bedrijf, waarin ongeveer twaalf man werkzaam zijn, heeft één winkel in de Amsterdamse Kalverstraat en een daarbij geïntegreerde webwinkel. Van hieruit verzendt every.day.counts naar 25 Europese landen. De collectie bestaat uit basisartikelen ('everyday') en 'statement pieces' ('today') die samen voor de dagelijkse outfit zorgen.

Voor every.day.counts ging expertgroeplid Maarten van Duijn op zoek naar een nieuwe ICT-architectuur. Het uitgangspunt bij de zoektocht was "zoveel als mogelijk binnen één geïntegreerd systeem". Het betrof een complete 'greenfield'-situatie, alleen de financiële consolidatie en de inkoop dienden in het ERP-systeem van het moederbedrijf plaats te vinden.

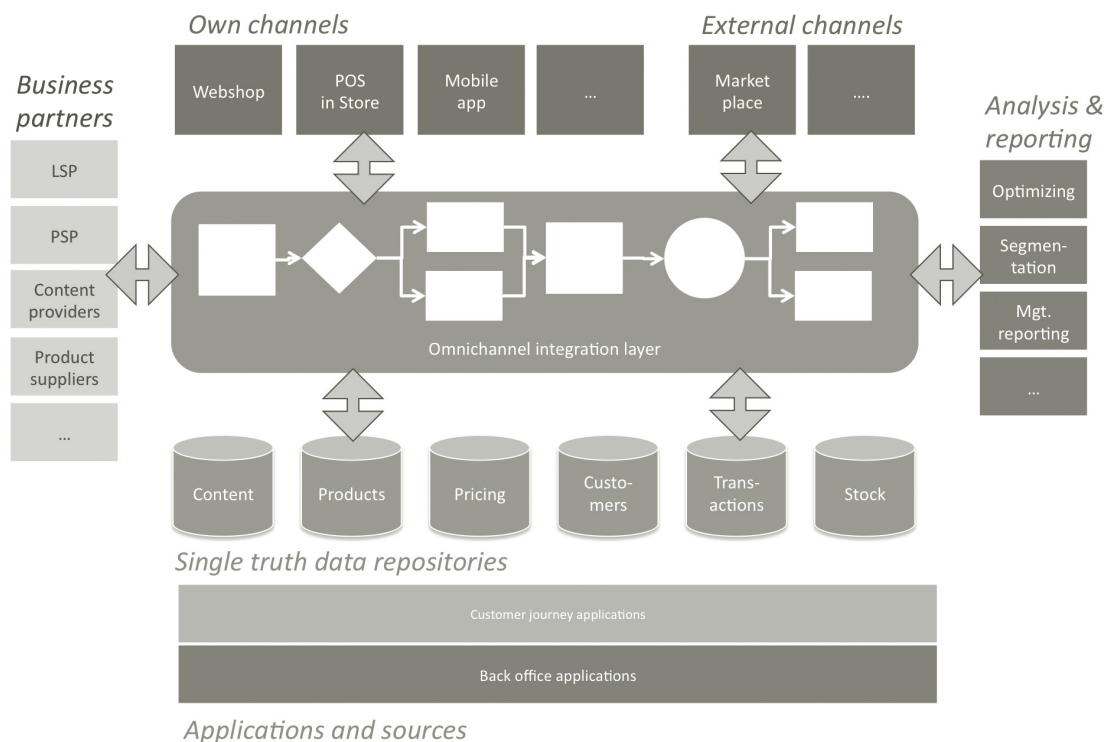
Maarten van Duijn over het resultaat: "Uiteindelijk bleek geen van de leveranciers op de shortlist een compleet geïntegreerd systeem te kunnen leveren. We zijn uitgekomen op een landschap waarin gebruikt wordt gemaakt van separate systemen voor webshop, kassa, ERP en digital asset management (PIM/DAM). Ik denk wel dat we met een groter budget meer functionaliteit in een meer geavanceerd systeem hadden kunnen onderbrengen."



Maarten van Duijn
IT Manager, Esprit

Een best-of-breed systeemlandschap kan vereenvoudigd worden weergegeven. Op hoofdlijnen zijn een aantal 'domeinen' binnen het landschap te onderscheiden:

- **Kanalen:** alle directe interactie met de consument verloopt via een aantal interne kanalen (verkoopkanalen, wijze van communicatie en gebruikte devices) zoals de webshop, kassa en app, maar ook via externe kanalen zoals een shop-in-shop op een internationale marktplaats
- **Business partners:** alle (integratie met) derde partijen die bijvoorbeeld product vendor, contentprovider en/of serviceprovider zijn. Denk hierbij aan de logistiek dienstverlener of payment provider
- **Omnichannel integratielaag:** in feite de 'dirigentfunctie' voor alle customer journeys en ondersteunende processen. Als ik een online verkoop in de winkel retour ontvang, waar moet het product dan naar toe, wat is dan het terug te geven bedrag, hoeveel loyaltypunten moet ik afboeken en had de klant eigenlijk al wel betaald? Deze 'laag' verzorgt tevens alle integratie tussen partijen en systemen (interfacemanagement)
- **Analyzing & reporting:** op basis van alle beschikbare, relevante gegevens (vaak uit combinaties van bronnen en systemen) worden analyses en rapportages gemaakt
- **Single truth data repositories:** de hierboven beschreven integratie laag en aanpalende systemen en partijen dienen allemaal gebruik te maken van dezelfde gegevens. Voor alle typen data dient dus een 'single truth' bepaald te worden. Vanuit die single truth repository worden vervolgens alle gebruikte systemen en partijen van data (de enige waarheid) voorzien
- **Applicaties en bronnen:** onderliggend wordt voor zowel functionaliteit als data gebruikgemaakt van (customer journey- en backoffice-)applicaties, die functionele domeinen voorzien van de juiste systeemondersteuning en data.



Conceptuele weergave van een omnichannel architectuur

In de praktijk moeten de bovenstaande rollen en de eerder genoemde functionaliteiten verdeeld worden over de verschillende software-oplossingen in het landschap. Welke software-oplossingen het beste zijn, verschilt per bedrijf. Het verdient echter wel de aanbeveling om bij het inrichten en beheren van een omnichannel architectuur een aantal ontwerpprincipes te hanteren. Deze werpen op de lange termijn hun vruchten af, maar zijn niet altijd de snelste of goedkoopste oplossing op de korte termijn.

5. Ontwerpprincipes

Voor het ontwerpen, inrichten en beheren van een omnichannel architectuur gelden de volgende ontwerpprincipes:

- Bouw de architectuur op uitlosse domeinen met autonome functies ('modulariteit'). Dit geldt ook als meerdere domeinen door dezelfde softwareoplossing worden ondersteund: dan dient deze software modulair te zijn ingericht. Dit vergroot de flexibiliteit en beheersbaarheid. Bovendien is het voor bepaalde functies belangrijk dat deze autonoom kunnen functioneren als (tijdelijk) niet met aanpalende systemen kan worden gecommuniceerd
- Beschouw de kanalen zoveel mogelijk als 'dunne' front-end-applicaties en breng er niet te veel complexe logica in aan. Houd ze daarmee flexibel voor wisselende content en interactie. Het 'dun houden' heeft bovendien veelal een positief effect op de performance
- Breng complexe systeem-overstijgende business-logica zoveel mogelijk onder in een flexibele 'tussenlaag'. Voor business-logica geldt ook vaak dat deze door leereffecten of veranderende marktomstandigheden aangepast moet worden
- Voldoe aan industriestandaarden op het gebied van:
 - o technieken en protocollen (REST, SOAP, Webservices, http(s), (s)FTP);
 - o berichtstructuren (marktstandaarden, techniekstandaarden zoals standaard Idocs, Bapi's);
 - o (product)coderingen en kenmerken (GS1-standaarden).Door hieraan te voldoen wordt flexibiliteit in integratie en samenwerking met business partners bereikt
- Hanteer het 'single source'-principe voor (master)data. Wanneer het repliceren van data noodzakelijk is, werk dan in een duidelijke master/slave-constructie
- Doorgrond het datamodel van toegepaste software-oplossingen en breng dit zoveel mogelijk in lijn voor alle applicaties. Dat vergemakkelijkt de integratie en eenduidigheid van data
- Borg de toegang tot de data(base) bij cloudoplossingen en Software-As-A-Service. Bedenk hierbij dat er een wezenlijk verschil is tussen het kunnen inzien

en/of aanroepen van data en het werkelijk kunnen beschikken over alle data, bijvoorbeeld om deze te combineren met data uit andere systemen.

6. Integratie

In een best-of-breed omgeving is een goede integratie tussen de verschillende software-oplossingen van groot belang. Dit wordt binnen de systeemarchitectuur meestal opgelost met een aparte 'omnichannel integratielaag'.

WE Fashion

Met de vernieuwde webshop wefashion.nl zet de organisatie in op een omnichannel retailstrategie. Naast de website maken hier ook mobiele apps en de kassaomgeving onderdeel van uit. Om een juiste omnichannel klantbeleving te bieden, ziet WE Fashion in dat een geïntegreerd landschap rondom de online en offline processen belangrijk is.

Om alle verschillende kanalen te verbinden, realiseerde WE Fashion een omnichannel integratielaag, genaamd SWITCH. Deze laag coördineert en koppelt alle primaire retailprocessen: denk hierbij aan processen rondom productdata, klant, prijs, voorraad, orders en retouren. WE Fashion kan hiermee snel inspelen op de internationale wensen en de diverse ontwikkelingen in de verschillende kanalen.

Over de praktische voordelen voor klanten zegt Geert Westendorp het volgende: "Klanten kunnen online kleding bestellen en het vervolgens afhalen in de winkel. Of in de winkel bestellen en laten leveren op een locatie naar voorkeur van de klant." Dit zijn voorbeelden van zichtbare veranderingen voor de klant dankzij de door WE Fashion gebruikte SWITCH-integratielaag.



Geert Westendorp
Manager Information Services, WE Fashion

Naast de keuze voor een software-oplossing dienen ontwerpkeuzes in de wijze van integratie gemaakt te worden. Binnen de huidige omnichannel architecturen is dit een van de meest cruciale aspecten. Voordat we dieper ingaan op de omnichannel integratielaag, staan we eerst stil bij de toekomst van integratie en bij de hedendaagse integratie-ontwerpkeuzes voor de belangrijkste gegevenstypen.

7. Vergezicht ten aanzien van integratie

In een ideale architectuur wordt alle data slechts eenmalig vastgelegd en ook slechts op één plek opgeslagen. Ervan uitgaande dat niet alle benodigde functionaliteit binnen een enkele applicatie kan worden geboden, betekent dit dat meerdere applicaties op een enkele database zouden moeten werken. Dit vraagt om een hoge mate van standaardisatie en openheid van applicaties, die in de huidige softwaremarkt nog niet toereikend worden ondersteund tussen verschillende applicaties.

Als voorbeeld zou dan het ERP-systeem van softwaremerk X gebruikmaken van een en dezelfde klantendatabase als het CRM-systeem van softwaremerk Y. Deze architectuur wordt al wel zichtbaar bij grote softwarepartijen, waarbinnen de verschillende modules en applicaties van die software op een en dezelfde database werken.

Qua technologie zal dit concept waarschijnlijk wel haalbaar blijken of al zijn. In hoeverre dit ook haalbaar is 'over softwarepartijen heen' hangt vooral af van andere aspecten die vaker een rol spelen bij standaardisatie, zoals organisatorische aspecten, machtsposities en commerciële belangen, bewaking van consistentie en internationale afspraken.

De 'second-best' architectuur is die waarbij applicaties webservices (diensten) van andere applicaties aanroepen om bepaalde gegevens te verkrijgen. Op die manier roept een webshop-applicatie de logistieke applicatie aan om de actuele voorraadstand van een product te verkrijgen zodra een consument dat product in zijn winkelmandje plaatst. Voor deze wijze van integratie geldt dat die over het algemeen goed toepasbaar is bij beperkte hoeveelheden data (transactie-gebaseerd, dus wel de voorraadstand van enkele producten, maar niet de voorraadstanden van alle producten als de consument een pagina met tientallen zoekresultaten oproept of door een catalogus aan het navigeren is).

In het geval van grotere hoeveelheden data is met de hedendaagse technologie een batch-gewijze integratie meestal de beste oplossing. De webshop ontvangt dan periodiek alle actuele voorraadstanden en gebruikt deze gegevens voor het tonen van beschikbaarheid. Pas wanneer een enkel item in de winkelmand wordt geplaatst, wordt de webservice met actuele voorraadstanden realtime aangeroepen voor een finale beschikbaarheidscontrole.

Voor beide situaties geldt dat de integratie vereenvoudigd wordt als de verschillende systemen hetzelfde datamodel hanteren. Denk hierbij bijvoorbeeld aan de wijze waarop producten en productvarianten aan elkaar gerelateerd zijn, hoe een systeem omgaat met bedrijven met meerdere vestigingen of dat per product gegevens op landniveau kunnen verschillen. Hoe meer gelijkvormigheid hierin is bereikt, hoe makkelijker de integratie en 'mapping' van berichtenverkeer verloopt.

8. Wijze van integratie voor de belangrijkste typen gegevens



EcommerceWiki/
ShoppingTomorrow

De beste wijze van integratie kan per gegevenstype (product-, prijs-, voorraad-, klant- en transactiedata) en voor management en informatieanalyse verschillen. Via EcommerceWiki.org kun je de volledige blueprint downloaden waarin in detail wordt ingegaan op de verschillende integratiescenario's per gegevenstype.

9. De omnichannel integratielaag

De omnichannel integratielaag vervult een aantal functies, die als volgt zijn op te delen:

- Uitvoeren en coördineren van processen en business rules over de verschillende applicaties
- Uitwisselen van informatie tussen de verschillende applicaties
- Inzien, muteren en beheren van informatie.

Om deze verschillende functies duidelijk te maken nemen we als voorbeeld een retourproces waarbij een online besteld product in een fysieke winkel wordt teruggebracht.

9.1 Uitvoeren en coördineren van processen en business rules over de verschillende applicaties

Om het product retour te kunnen nemen, dient de kassa de originele online verkooporder op te roepen. Hiervoor maakt deze gebruik van de omnichannel integratielaag. Deze geeft op basis van de originele weborder het exacte, terug te geven bedrag voor het individuele item, ook wanneer op de gehele order een korting van toepassing was. Op basis van business-logica bepaalt de omnichannel integratielaag of ook de transportkosten van de thuisbezorging teruggeven moeten worden. Dat is namelijk bij 'gold members' niet het geval. Wel dient het saldo spaarpunten verminderd te worden. Hiertoe muteert de omnichannel integratielaag in het loyalty-systeem.

Door dergelijke processen en business rules te modelleren in de integratielaag kunnen betrouwbare, flexibele en herbruikbare oplossingen worden gerealiseerd. De integratielaag (vaak Enterprise Service Bus, of kortweg ESB genoemd) coördineert de verwerking van deze processen die door de verschillende kanalen worden toegepast.

9.2 Uitwisselen van informatie tussen de verschillende applicaties

In bovenstaand voorbeeld moeten waarschijnlijk een aantal verschillende systemen aangeroepen, gemuteerd en/of geïnformeerd worden. Hierbij kan het ook om systemen van derde partijen gaan, bijvoorbeeld een externe loyalty card-partner.

Elk systeem of elke partner kent veelal zijn eigen berichtstructuur of voorwaarden. De ESB is verantwoordelijk voor het uitwisselen van de informatie tussen deze kanalen. De bron van deze informatie bestaat veelal ook uit verschillende applicaties waardoor er een oplossing nodig is om berichten te vertalen en betrouwbaar uit te wisselen. De ESB voorziet hierin door:

- berichtschema's te ondersteunen van de verschillende applicaties en systemen;
- deze berichtschema's te vertalen (mappen) van het ene bericht naar het andere;
- veelal bij deze vertaling informatie te combineren of te 'verrijken', zoals het combineren van productinformatie met prijs- en voorraad informatie;
- de berichten volgens een gedefinieerd en gecontroleerd proces uit te wisselen en deze in verschillende transportmethodes aan te bieden (waaronder FTP, Webservices, HTTP-s, REST, Adapters als BAPI's en EDI);
- het berichtenverkeer te beheren door toepassen van queuing, validaties, monitoring en rapportages.

In bovenstaand voorbeeld moet de terugbetaling aan de consument ook kunnen plaatsvinden wanneer het loyalty-systeem onverhoopt niet beschikbaar is. De ESB zal deze mutatie dan 'vasthouden' en op een later moment alsnog doorvoeren.

9.3 Inzien, muteren en beheren van informatie

In veel gevallen is de ESB het enige systeem dat informatie kan verschaffen over hoe een proces exact verlopen is en welke gegevens gebruikt zijn. Daarmee is het voor een klantenservice, de winkelmedewerkers maar ook de afdeling ICT een hele bruikbare bron van informatie. Toch is het ook vaak gewenst de mogelijkheid te hebben om processen te sturen en/of gegevens te muteren. Als voorbeelden:

- Een klant heeft online een order besteld maar wenst de leverdatum aan te passen en belt hiervoor de klantenservice
- De klant heeft een retour opgestuurd maar is vergeten de aankoopnota mee te sturen. De retour is ontvangen en moet gekoppeld worden aan een order na een telefoontje met de klantenservice
- De klant belt met een klacht omdat een promotie actief is en toch blijkt dat de aankoop in de winkel geen rekening gehouden heeft met eerdere aankopen. De klantenservice wenst de klant hiervoor een correctie of promotie te sturen.

Voor dit soort toepassingen dient de ESB voorzien te zijn van gebruikersschermen en moet deze het muteren van gegevens door meerdere gebruikers goed en veilig ondersteunen. Met name dit aspect maakt dat een 'traditioneel' middleware-

pakket niet meer volstaat, of uitgebreid dan wel aangevuld moet worden met deze functionele gebruikersinteractie.

10. Conclusies

Een optimale omnichannel klantervaring én het optimaliseren van de winstgevendheid van een retailoperatie vragen om veel functionaliteit. Hoeveel functionaliteit je ook binnen één platform kunt onderbrengen, het totale systeemlandschap bestaat altijd uit meerdere applicaties ('best of breed'). Ook bij gebruik van moderne technologieën blijft integratie een onderwerp dat veel aandacht vraagt.

Voor ons zijn de volgende zaken daarbij de belangrijkste conclusies uit de kennisdeling die binnen ShoppingTomorrow heeft plaatsgevonden:

- Bouw de architectuur op uit losse domeinen met autonome functies ('modulariteit')
- Beschouw de kanalen zoveel mogelijk als 'dunne' front-end-applicaties en breng er niet te veel complexe logica in aan
- Breng complexe systeemoverstijgende business-logica zoveel mogelijk onder in een flexibele 'tussenlaag'
- Voldoe aan industrie-standaarden (markstandaarden, technologiemarkstandaarden)
- Hanteer het 'single source'-principe voor (master)data
- Doorgrond het datamodel van toegepaste software-oplossingen en breng dit zoveel mogelijk in lijn voor alle applicaties
- Borg de toegang tot de data(base) bij cloudoplossingen en Software-As-A-Service.

"Zo moeilijk kan het toch niet zijn?" is een veelgehoorde kreet als het gaat om nieuwe eisen en wensen aan het omnichannel ICT-landschap. Wij kunnen niet anders dan concluderen dat het bieden van een perfecte, naadloze en geoptimaliseerde omnichannel klantervaring een complexe aangelegenheid is, zeker als deze met een mix van nieuwe en bestaande systemen moet worden gerealiseerd.



Meer lezen?

Ga naar EcommerceWiki.org om de volledige blueprint van de expertgroep Omnichannel Architectuur te downloaden.

*EcommerceWiki/
ShoppingTomorrow*

GASTHEER



Aroen Sharma
Director Solution Engineer
SAP

VOORZITTERS



Andre Damsteegt
Directeur
Magnus



Axel Groothuis
Partner
Magnus

Leden expertgroep



Maarten van Duijn
IT Manager
Esprit



Robin Gabriner
Consultant
Magnus



Roland van Kortenhop
Manager Operations
Thuiswinkel.org



Geert Westendorp
Manager Information Services
WE Fashion



Nico Wartenbergh
CIO
Rituals



Roel Brand
IT Manager
Homefashion Group



Willem Verwijs
Manager ICT
BeterBed



Jeroen van de Bruele
Manager IT
Shoebly



Guus Brinkel
Head of IT
Hunkemöller



Michel Ruijterman
Senior Director IT Online
Ahold



Simon Hansen
CIO
Intergamma



Anjo van der Spek
CIO
DMG / De Mandemakers
Groep