

# Adobe Experience Manager

## Onderzoeksvoorstel Bachelorproef

Mike Heymans<sup>1</sup>, Steven Rymenans<sup>2</sup>

### Samenvatting

Adobe Experience Manager (AEM) is een content management applicatie van Adobe dat een bedrijf in staat stelt om zijn web content te bewerken zonder tussenkomst van IT. Designers of managers kunnen zelf pagina's aanpassen en opnieuw publiceren. Dit is ideaal voor statische data of specifieke content maar wat met bv. prijzen van een webwinkel? Het is onmogelijk om alle product pagina's manueel aan te passen tijdens de solden, dit vereist een andere aanpak. Er zijn 2 manieren om AEM van deze dynamische data te voorzien, de eerste is dat bij elke wijziging de data naar AEM wordt verzonden zodat deze ze opnieuw kan aanmaken, publiceren en eventueel cachen. Het grote voordeel hiervan is dat er geen connectie naar de database moet gebeuren wanneer een bezoeker de pagina opvraagt maar de keerzijde van deze methode is dat het genereren van de pagina's een intensieve operatie is wat bij grote, wisselvallige data tot overbelasting kan leiden. De 2de manier is via rest calls de data ophalen wanneer de gebruiker deze nodig heeft wat op zich een vertraging in weergave kan voortbrengen. Maar waar trekken we de grens tussen dynamische en statische data? Wat zijn de voorwaarden waaraan data moet voldoen om de pagina's te laten regenereren en wat komt er binnen via injectie? Deze proef dient om een antwoord op deze vragen te vinden en een optimalisatie plan voor het AEM-platform. Hieruit kunnen we afleiden wat de beste manier is om bepaalde data naar content om te zetten zonder risico voor overbelasting. Dit document zal voor toekomstige AEM-projecten een handleiding zijn waarbij AEM uit te doeken wordt gedaan. De handleiding zal developers begeleiden bij de opzet van het platform alsook richtlijnen voorzien over het omgaan met data. Hierbij kan de gebruiker terugvallen op de simulaties die we in dit onderzoek zullen voltrekken, zodat hij of zij van in het begin de juiste aanpak kiest en zichzelf een hoop ellende bespaart.

### Sleutelwoorden

Webapplicatieontwikkeling. Adobe Experience Manager — Microservices — CMS — Marketing

Contact: <sup>1</sup> mike.heyman.v7773@student.hogent.be; <sup>2</sup> steven.rymenans@asadventure.com

## Inhoudsopgave

1	Introductie	1
2	State-of-the-art	1
3	Methodologie	2
4	Verwachte resultaten	2
5	Verwachte conclusies	2

publiceren? Adobe Experience Manager biedt deze feature waarbij razendsnel content kan aangepast worden op een live site en dit niet enkel door ITers. Een html-leek kan perfect een nieuwe pagina aanmaken, voorzien van de nodige tekst en foto's om vervolgens deze live zetten. Natuurlijk kan niet alles via deze manier gebeuren en zal een deel van de data vanuit een databank opgehaald moeten worden. Dit is iets lastiger en vereist een plan van aanpak.

## 1. Introductie

Tegenwoordig wordt er van websites verwacht dat ze snel en dynamisch kunnen inspelen op actuele gebeurtenissen door hun content van gerelateerde inhoud te voorzien. Vroeger gebeurde dit door de inzet van een developer maar dit zorgt voor een onoverkomelijke vertraging tussen een manager die het concept bedenkt en een developer die het uitvoert. En na realisatie kan de manager ontevreden zijn over het resultaat waardoor het concept terug bij de developer moet passeren waarna het hopelijk wel gepubliceerd kan worden. Dit process kan enige tijd in beslag nemen waardoor het moeilijk/onmogelijk is om snel in te spelen op events maar wat als de manager zelf de content kan aanpassen, goedkeuren en

## 2. State-of-the-art

Momenteel werk ik mee aan een project waar AEM instaat voor de content van een online shop. De website wordt op 2 manieren voorzien van data. De eerste manieren is voor data die zelden wijzigt zoals de omschrijving van een product of de openingsuren van een winkel. Deze wordt op moment van wijziging naar AEM gestuurd, waarop deze de gewijzigde pagina's aanpast en opnieuw publiceert. De tweede manier is voor variabele data zoals prijzen of de leveringstermijn van producten. Die data worden via rest-services opgehaald op het moment dat een gebruiker deze nodig heeft. Het is al voorgekomen dat we te veel data via de eerste manier naar AEM duwen waardoor deze machines het begeven en alle

content generatie stilvalt. Ondanks onze ervaring met het platform hebben we nog altijd geen richtlijnen i.v.m. het beheren van de data.

### 3. Methodologie

Om te achterhalen wat de beste manier van aanpak is i.v.m. data management binnen AEM, ben ik van plan om zelf een minimalistische AEM stack op te zetten. Aangezien AEM bekend staat om zijn content management gaan we ons vooral toeleggen op het automatisch laten genereren van frontend, gesitueerd in een e-commerce omgeving. Via een rest service wordt er dan periodiek een databank aangesproken om wijzigingen naar het AEM-platform te verzenden. We beginnen bij een kleine wijziging van records (een honderd tal) en vergroten deze waarbij we de tijd monitoren dat AEM nodig heeft voor het repliceren van de pagina's. Wanneer de tijd een op vooraf afgesproken norm overschrijdt, kunnen we concrete voorwaarden leggen op data omtrent zijn veranderlijkheid.

### 4. Verwachte resultaten

Een concreet set richtlijnen waaruit we kunnen afleiden wat de optimale manier van content management is via het AEM platform, gebaseerd op de veranderlijkheid en grote van de datasets. Hierdoor zal op voorhand voor de juiste aanpak gekozen kunnen worden i.p.v de huidige trial-and-error methodology. Dit, samen met een stap-voor-stap uitleg over de opzet van een AEM-applicatie, maakt dat een beginnening vol zelfvertrouwen aan zijn eerste AEM ervaring kan beginnen.

### 5. Verwachte conclusies

Het is te verwachten dat kleine, specifieke wijzigingen van een beperkte dataset manueel kan gebeuren (zoals de layout van winkelpagina's). Redelijk statisch data (zelden meer als één wijziging per dag) kan op moment van wijziging naar AEM gezonden worden waarop deze de pagina opnieuw kan genereren en publiceren. De constant veranderende dataset, die van minuut tot minuut kan verschillen, zal last-minute via rest-services worden opgehaald en geïnjecteerd in een template pagina.