

## Reflectie: web- and native applications

Jasper D'haene  
Florian Dejonckheere

## Native Applications and Their Successor: Web Applications

De opkomst van de smartphone zorgde voor een revolutie op het vlak van mobiele dataconsumptie. De informatie wordt daarbij verwerkt op een relatief krachtig mobiel toestel die praktisch altijd toegang heeft tot een dataverbinding. Daardoor kan hybride gewisseld worden tussen snellere maar lokale verbindingen zoals Wi-Fi, en tragere, globale GSM-netwerken.

Applicaties kunnen grofweg opgedeeld worden in twee grote groepen: de *web applications* en de *native applications*. Daarbij is het grootste verschil tussen de twee dat web-applicaties in de regel minder snel zijn, en dus minder geschikt voor rekenintensieve taken. Daarvoor zijn native apps dan wel weer bekwaam.

Ook voor de ontwikkelaar is het verschil tussen de twee pijnlijk duidelijk. Applicaties die op het web draaien – *in de cloud* – zijn op zich gemakkelijker te schrijven vanwege het brede scala aan apparaten waarop deze kunnen draaien, waardoor de codebase relatief klein blijft. Hierbij moet natuurlijk ook de recente *App Runtime for Chrome* genoemd worden, die bepaalde native Android-apps op het Chrome-platform uitvoert.

Deze divergentie van apparaattechnologie brengt natuurlijk ook zijn nadelen mee, met name backwards en forwards compatibiliteit, en een overvloed aan ondersteunde randgevallen. Ook het onderhoud van webapps is eenvoudiger – er hoeft slechts één instantie (op gerepliceerde systemen na) geüpgraded worden, in tegenstelling tot uitgebreide en complexe deploymentsystemen zoals *Google Play* en de *Windows Store*.

Native apps hebben dan weer het voordeel van integratie. Deze beschikken over een grotere controle van het systeem, waardoor mogelijkheden zoals toegang tot randapparaten en OS-level functionaliteit mogelijk is. Deze beperking wordt creatief omzeild door het hybride ecosysteem *Firefox OS*, waarbij de user interface én alle apps geschreven zijn in HTML, CSS en JavaScript.

Maar ook grotere systemen – tablets, laptops en dergelijke – kunnen profiteren van het verschil. In casu zijn de Google Chromebooks niet meer dan flat clients die webapplicaties draaien, in plaats van de traditionele application stack.

Uiteindelijk is de keuze van platform afhankelijk van het functionele aspect van de applicatie. Als die onafhankelijk kan of moet draaien – dus zonder dataverbinding – kan de native aanpak eerder naar voren geschoven worden. Als men echter een applicatie ontwikkelt die afhankelijk is van een webservice, kan een webarchitectuur een interessantere keuze zijn.