

Koeien & Java

Java technologie in de veehouderij

Samen coderen met veehouders, proberen het vak van een veehouder goed te begrijpen en natuurlijk heel veel plezier met Java development. Dat is kort samengevat ons werk bij een coöperatie voor veehouders, waar we met een team van First8 zitten. We vinden het leuk om over ons vak te vertellen: van innovatieve Java development tot samenwerking met de veehouders. Na een korte inleiding over het bedrijf en de veehouderijbranche vertellen we over de Java technologieën, die we er toepassen.

Interessante domeinkennis

De Nederlandse agrarische sector heeft een miljarden omzet en maakt volop gebruik van automatisering. Onze klant CRV levert als coöperatie voor melkveehouders vele IT-diensten op het gebied van veemanagement. De veehouders sturen via deze diensten de boerderij aan. Het mooie aan het IT-vakgebied is dat elk project weer een eigen domein heeft. Als je projecten hebt gedraaid in de financiële sector, bij mediabedrijven, in de logistiek of waar dan ook, dan is de verleiding groot om te denken: hoe complex kan het zijn om een boerderij te runnen? En toch worden we telkens verrast door de unieke uitdagingen binnen dit vakgebied. Gelukkig helpt het boerenverstand!

Als je kijkt op de site www.CRV4all.nl, dan zie je meteen dat het boerenbedrijf veel meer behelst, dan alleen het hebben van dieren. Het gaat bijvoorbeeld om topstieren, goede voedselkwaliteit en efficiënt melken. De moderne boerderij is een onderneming met de boer als ondernemer aan het stuur, vaak met personeel in dienst. Nederland staat bekend om zijn voedselkwaliteit en er is internationaal veel vraag naar Nederlandse melk, kaas en vlees. Veehouders moeten dan ook op vele gebieden voldoen aan strenge normen (wetgevingen, op gebied van hygiëne, voedselkwaliteit en mest) en hebben daarom veel informatie nodig. Hiervoor maken wij gerichte oplossingen, zoals rekenmodules om de optimale verwerking van mest te berekenen. Centraal staat de veehouder en het verschaffen van inzicht voor een optimale veestapel.

Drie concrete voorbeelden

Als ingelogde veehouder krijg je via de website veel inzicht in je veestapel. Bijvoorbeeld: wat geef je de koeien wanneer te eten? De soft-

ware verschaft overzicht, afhankelijk van je productiedoelen en de situatie. En voedsel dat erin gaat, komt er ook weer uit. Dezelfde data wordt dus doorberekend in het mestbeleid, om te voorkomen dat de hoeveelheid mest buiten de wettelijke kaders gaat. Voorbeeld 2: welke stieren wil je voor de toekomst van je veestapel? Een webshop biedt een keuze uit topstieren met tienduizenden nakomelingen vanwege hun krachtige erfelijke eigenschappen. Is het de eerste keer voor de koe, dan is het wellicht beter om een stier te kiezen met een geboortegemak indicatie. Men kan natuurlijk ook optimaliseren voor melkproductie of levensduur. Dan heeft het zin om eerst onderzoek te doen via onze software. Voorbeeld 3: hoe kom je erachter of een koe überhaupt bevrucht kan worden? Dat kan door het diergedrag te observeren, iets dat bij een groot bedrijf niet altijd praktisch is. Moderne bedrijven hebben hierom stappentellers, die allerlei stalinfo opleveren. Hieronder ook de informatie of een koe bevrucht kan worden. Dit zijn gegevens die via de site worden ontsloten. Internet of Things is overigens al ver doorgedrongen op de boerderij: van Fitbits voor koeien tot melkrobots.

Uitdaging voor de ICT

De organisatie staat voor veel (herkenbare) uitdagingen in het software landschap. Jaren geleden is gekozen voor een betaald ontwikkelpakket. In de loop der jaren is duidelijk geworden, dat er veel behoefte is aan een nieuwe, open architectuur. De coöperatie wil allereerst ook (internationaal) groeien en men kiest daarom voor een nieuwe, schaalbare infrastructuur. De klanten komen met steeds meer verzoeken om mobiele applicaties en verwachten moderne webapplicaties. De markt eist steeds meer wendbaarheid, meer dan het commerciële pak-



Arjan Lamers
is open source
software architect
bij First8 BV



Koen Aben is
Senior Grails en
Java specialist bij
First8 BV

ket kon leveren. Daarom is er gekozen voor Grails als platform voor het snel ontwikkelen van webapplicaties. Hiermee wordt op termijn de hele voorkant vervangen. Tegelijkertijd wordt de achterkant vernieuwd. Op basis van de open source stack van Red Hat wordt de backend nagebouwd met een moderne microservices architectuur, die weer een flinke tijd mee moet kunnen gaan.

Scrum

Men gebruikt al jaren Scrum als methode voor softwareontwikkeling. De Sprints zijn telkens twee weken, dat zorgt voor focus en snelle feedback. Het goede is dat ook niet-development partijen (zoals klantenservice, beheer en marketing) heel goed bekend zijn met de Scrum-methode en proactief meedenken. Zo blijven de ontwikkelteams goed doordrongen van de markt wensen en kan men begripvol schakelen met beheer voor productie deployments. De Sprint demo's worden doorgaans online bijgewoond door een drietal boeren, die zorgen voor directe feedback en voor extra focus op ieders boerenverstand.

De veehouderij is zeer omvangrijk qua domeinkennis. Dankzij de multifunctionele teams gaat de kennisoverdracht vanzelf. De domeinexperts in elk Scrum team weten bijvoorbeeld als geen ander welke nieuwe wetten er in 2016 aankomen en wat dit voor de achterban betekent. Deze domeinexperts, vaak parttime veehouder, helpen direct mee aan de ontwikkeling van de features. Zo leer je als ontwikkelaar te denken als een veehouder. Erg leuk, maar niet eenvoudig. Er komt namelijk veel jargon en branche ervaring bij kijken. Gelukkig is elk team multidisciplinair: de teamleden zijn een mix van programmeurs van de oorspronkelijke systemen, domeinexperts en experts op het gebied van Grails.

Groovy en Grails

Grails is een framework voor de JVM bedoeld voor hogere productiviteit in webdevelopment, gebaseerd op Groovy en Spring. Met andere woorden: het heeft heel veel features, die het makkelijk maken om snel webfunctionali-



teit te bouwen. Met Grails worden vele zaken eenvoudiger, hieronder enkele voorbeelden:

- het bouwen van een REST API (XML of JSON);
- het bouwen van CRUD functionaliteit;
- integratie met Spring frameworks (zoals Spring Security);
- integratie met andere frameworks (zoals Hibernate, Quartz, Less CSS, etc.).

Deze zaken zijn makkelijker, omdat Grails een abstractielaag legt boven deze functionaliteit. Het standaardiseert veelvoorkomende zaken in webapplicaties. Dit maakt Grails erg krachtig om te gebruiken, maar het eist wel veel gedegen vakkennis van zowel dit slimme framework als de onderliggende Java stack.

Grails is allereerst en hoofdzakelijk een effectief webapplicatie framework, maar het is ook een uitbreidbaar platform. Het platform heeft dankzij een plug-in mechanisme krachtige extension points, die uitbreidingen toestaan, zelfs op het gebied van de runtime engine.

Omdat we standaard heel veel in plug-ins stoppen (data models, resources, standaard componenten), is een krachtig basisplatform ontstaan voor onze Grails applicaties. Hierdoor kunnen we makkelijker functionaliteit hergebruiken, kunnen we sneller applicaties ontwikkelen en kan men nieuwe applicaties direct laten conformeren aan de huisstijl.

Inmiddels zijn er vele applicaties gebouwd (of herbouwd) in Grails volgens de Scrum werkwijze. Grails maakt het mogelijk om elke twee weken nieuwe features te ontwikkelen voor onze eindgebruikers, zoals ook door Scrum wordt beoogd. Zo werd CRV onlangs de eerste leverancier in de branche met software voor het simuleren van mestverwerking volgens de nieuwe wetgeving. Deze module genaamd "Grondgebondenheid" werd gebouwd met eindgebruikers in het team. Dit zorgde voor directe feedback, een snelle oplevering en goede waardering van veehouders.

Microservices

Toen Grails werd geïntroduceerd, was Java nog nieuw in het softwarelandschap dat wij aantroffen. Inmiddels is de toekomstvisie volledig op de Java stack gericht. De reden ervoor is logisch. Men heeft zoveel services en applicaties voor de agrarische sector, dat dit vraagt om een hele open en flexibele stack van tools voor visualisatie, processing en beheer van services. Gekozen is voor een open source (Red Hat) microservice architectuur.

De bestaande, monolithische backend wordt functioneel opgedeeld in *domeinen* en elk domein krijgt een strak gedefinieerde API. Daar kunnen de Grails applicaties dan van gebruik maken. Door de opsplitsing in domeinen kunnen de programmeurs en domeinexperts zich focussen op een specifiek deelgebied, zodat ze onafhankelijk van

de andere teams besluiten kunnen nemen en de benodigde domeinkennis beperkt kan blijven.

Bij elk domein wordt schaalbaarheid en performance constant in het achterhoofd gehouden. Niet alleen moet het platform veel data kunnen verwerken op wereldwijde schaal, maar ook moeten we rekening houden met de vaak slechte internetverbindingen in sommige landen. Daarbij wordt kritisch gekeken naar welke processen *snel* resultaat moeten opleveren en welke processen relatief langer op een antwoord kunnen wachten.

Om dit alles goed te kunnen doen, moeten de nieuwe domeinen dus goed geïsoleerd zijn. Dat betekent dat bestaande processen, die al jaren bestaan, kritisch bekeken worden en misschien opgeknipt worden. Dat vereist veel discipline en creativiteit van de domeinexperts om aan de eisen van de software architectuur te kunnen voldoen. Het is mooi om te zien dat de rollen af en toe eens omgedraaid worden. Vaak zijn het de domeinexperts, die de randvoorwaarden neerleggen. Hier wordt heel actief de discussie opgezocht.

Een mooi voorbeeld kwam naar boven bij het ontwikkelen van de *ubiquitous language*: een verzameling van termen waar iedereen hetzelfde idee bij heeft. Een overduidelijk onderdeel bij CRV is natuurlijk de *koe*, want daar draait het hele bedrijf immers om. Maar zo simpel is het niet. Er zijn stieren, kalveren, drachtigen, vaarsen, pinken, pinkenstieren, etcetera. En dan heb je nog de term 'koe' zelf, dat de diersoort beschrijft, maar ook het woord is voor een *vrouwelijke koe*. De meest belangrijke term in dit domein is dus verschrikkelijk ambigu.

De Stack

De voorkant van het softwarelandschap bestaat voornamelijk uit Grails 2 en Grails 3 applicaties. Hier is ruimte in diversiteit, omdat de lifetime van deze applicaties een stuk korter is; er wordt vooral snel en wendbaar ingespeeld op actuele marktwensen. Voor de datadomeinen hebben we een veel langere levensduur voor ogen. Daarom is ook gekozen voor een open source stack op basis van de aankomende Red Hat release (EAP 7): Wildfly 10. Dit wordt gecombineerd met Oracle en PostgreSQL.

De verscheidenheid aan applicaties, gecombineerd met een microservices architectuur, maakt het noodzakelijk om een gedegen single-sign-on (SSO) systeem te hebben. Daar komt nog eens bij dat steeds vaker externe identity

providers in het spel komen: van social login's (via Facebook en Google) tot en met overheidsdiensten als eHerkenning. We hebben overal mee te maken. Daarom hebben we gekozen voor Keycloak als identity manager en dat gecombineerd met Apiman. Beiden zijn producten waar Red Hat aan meewerkt en dat houdt onze stack overzichtelijk.

Gezien de schaalbaarheidseisen moeten we in staat zijn om snel nieuwe servers uit te rollen. Dat betekent dus dat *alles* geautomatiseerd wordt met behulp van Puppet: van het inrichten van databaseservers tot en met het provisionen van Wildfly en Keycloak. Een bijkomend voordeel is dat we nu met behulp van Vagrant met één druk op de knop een development omgeving uit de klei kunnen trekken.

Tot slot

Java speelt een cruciale rol in de agrarische sector. In welke sector niet eigenlijk? Java is everywhere. Elk domein heeft zijn eigen uitdagingen en wij kunnen iedereen van harte aanbevelen om regelmatig eens Java uit te proberen in een compleet andere sector. Onderschat nooit wat erbij komt kijken om wat "gewoon" boerenverstand te hebben. ■

**HOE COMPLEX
KAN HET ZIJN
OM EEN
BOERDERIJ
TE RUNNEN?**

