

Moving Java Forward

JavaOne 2011 - een terugblik naar de nabije toekomst

samenwerking tussen Luminus (Paul Bakker en Bert Ertman) en AMIS (Lucas Jellema)

Begin oktober was de week van de waarheid voor Java: de jaarlijkse JavaOne conferentie in San Francisco, onder de veelbetekenende titel: Moving Java Forward. Sinds ruim anderhalf jaar geleden Oracle de touwtjes van Sun in handen kreeg wordt deze conferentie gelijktijdig gehouden met Oracle's eigen Open World conferentie. Met ontegenzeggelijk logistieke voordelen (zoals een concert van Sting) maar ook uitdagingen voor wat betreft bijvoorbeeld het eigen karakter van JavaOne. Na het aftasten van afgelopen jaar moest deze editie uitwijzen of JavaOne op deze manier bestaansrecht heeft. En belangrijker nog: dit jaar moest bewijzen dat onder Oracle's 'stewardship' het Java Platform en de Java Community weer een opgaande lijn zouden gaan volgen, na jaren van stagnatie en zelfs achteruitgang. In dit artikel beschrijven Paul Bakker, Bert Ertman en Lucas Jellema - alledrie als spreker aanwezig op JavaOne - hoe zij de conferentie hebben ervaren en wat ze de belangrijkste conclusies vinden met betrekking tot de staat en toekomst van Java in alle aspecten.



Foto: Adam Messinger (VP Fusion Middleware) tijdens JavaOne Strategy Keynote

Om maar meteen met de grootste uitdaging te beginnen: Een van de voornaamste problemen die Oracle kende bij het overnemen van het stewardship van Java was de omgang met de developer community. Geen enkel ander softwareontwikkelpatform kent zo'n bepalende community als dat we die bij Java kennen. Als je daar onderdeel van uitmaakt is dat iets om heel trots op te zijn en te koesteren. Voor een leverancier die ergens zijn stempel op wil drukken is dat een hel. Toch heeft het succes van Java in de afgelopen zestien jaar alles te maken met hoe Sun het heeft losgelaten in de community. Zonder de community zou er nu geen Java zijn. Vanaf het moment van overname van Sun probeerde Oracle de community te besturen vanuit haar fieldmarketing divisie. Dezelfde manier zoals ook de Oracle product User Groups worden bestuurd. Immers: Java gebruikers waren (potentiele) 'klanten' van Oracle. De afgelopen anderhalf jaar hebben echter op een dramatisch wijze uitgewezen dat een developer community zich niet op een gelijkwaardige manier laat besturen en zeker niet door Oracle - een bedrijf dat onder veel Java ontwikkelaars een dubieuze reputatie van genadeloze 'casher' heeft. Het moest dus over een andere boeg en snel ook, want anders zou Oracle het laatste restje geloofwaardigheid als hoeder van Java gaan verliezen. Nadat er een storm van kritiek was

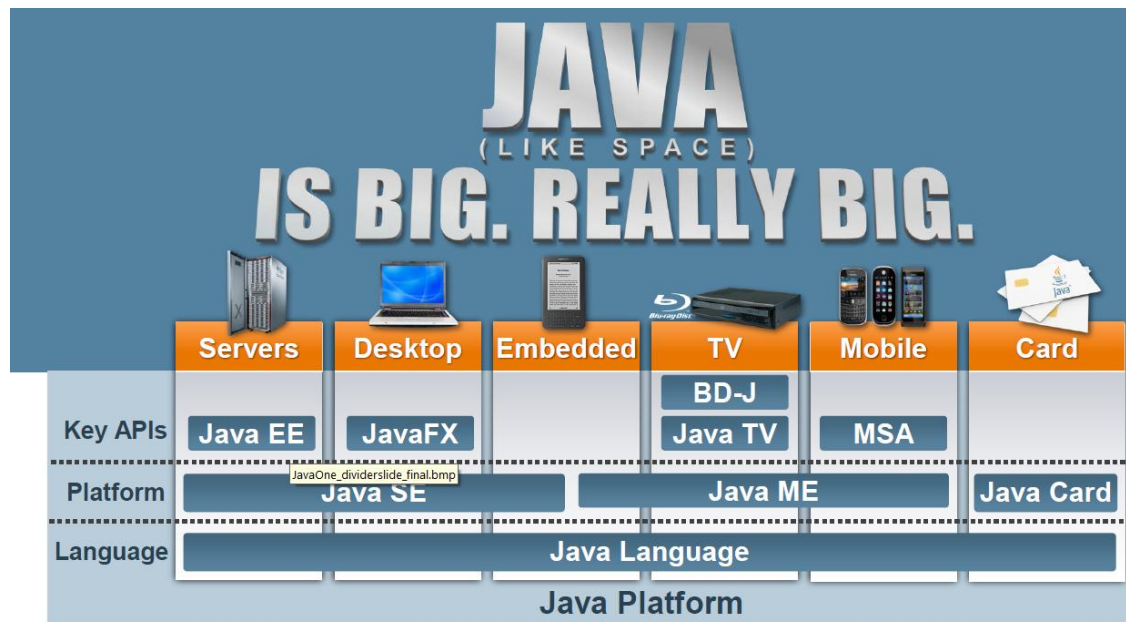
losgebarsten over de manier waarop Oracle met het Java platform om was gegaan en tevens hoe de developer community zich volledig buitengesloten voelde, was het signaal luid en duidelijk. Dit moest anders. Oracle heeft hiertoe besloten om op diverse fronten actie te ondernemen. De vertegenwoordigers van de Java community - de Java User Group Leaders en Java Champions - werden nadrukkelijker uitgenodigd bij allerlei strategie briefings. Ook werden er diverse maandelijkse conference calls ingesteld om zodoende meer vinger aan de pols te houden bij wat er leeft in de community. Deze openheid ten aanzien van het delen van de technische koers van Java was voor Oracle nieuw. Het was duidelijk dat ze hier enigszins ongemakkelijk bij waren getuige de vele sessies die alleen onder NDA plaatsvonden en de vele safe harbour statements in presentaties en hand outs.



Figuur: Moving Java Forward - platform, community en vendors

Desalniettemin werd langzaam een gedeelte van het verloren vertrouwen terug gewonnen. Niet in de laatste plaats omdat ook de organisatie van de JavaOne - jarenlang toch het vlaggenschip voor community en vooruitgang op het gebied van Java - open stond voor inspraak. Vanuit Nederland hebben we daaraan ook duidelijk ons steentje bijgedragen. Namens de NLJUG heeft Bert Breeman deelgenomen aan de tienkoppige steering committee van JavaOne en Bert Ertman was onderdeel van de program committee welke het programma van de conferentie mocht meebepalen. Andere belangrijke wapenfeiten waren de toetreding van twee belangrijke Java Users Groups (SouJava uit Brazilië en LJC uit Londen) tot de JCP Executive Committee. De erkenning van de development community is een zeer belangrijke overwinning voor ons als Java developers. Het is geen ordinaire machtsstrijd, maar een teken van het feit dat Java niet te 'productizen' is. Java is een technologisch ecosysteem waardoor naast Oracle ook andere bedrijven en individuele personen bestaansrecht hebben. Oracle is de steward van Java, maar

heeft niet alles alleen voor het zeggen. De input van anderen is belangrijk voor het voortbestaan van het platform. Dit geldt niet alleen voor individuele developers, maar ook voor het ecosysteem van de vendors die vertegenwoordigd zijn in de JCP. Het afgelopen jaar werd natuurlijk gekenmerkt door de rechtzaak tegen Google rondom Android en het verlaten van de JCP door onder andere Apache. Beide onderwerpen werden overigens in alle officiële uitingen rondom de conferentie handig omzeild of verzwegen. Hoewel je hier nog duidelijke tekenen ziet van de juridische afdeling die achter de schermen nog krampachtig probeert de touwtjes in handen te houden gebeuren er ook genoeg positieve dingen die maken dat vendors een positieve toekomst voor Java (blijven) zien. Met JCP.next zijn een set aan veranderaatregelen voorgesteld om de JCP meer transparantie te geven. Er komen weer nieuwe versies van het Java platform uit die al dan niet zonder slag of stoot door de JCP geratificeerd worden. Nieuwe partijen sluiten zich aan bij OpenJDK en de JCP, kortom het ecosysteem rondom Java heeft een stevige reanimatie nodig gehad. Na zo'n actie is het altijd nog maar bezien hoe groot de opgelopen schade is, maar het belangrijkste nieuws is dat de patiënt nog leeft en zijn weg kan vervolgen. Java beweegt weer voorwaarts en dat is een prima uitgangspunt om een conferentie als JavaOne mee te laten vertrekken.



Cloud

De belangrijkste trend was dit jaar duidelijk de cloud. Ondanks dat dit al een aantal jaar een belangrijk onderwerp is, is het nu iets wat ook echt gedaan wordt in plaats van dat er alleen maar over gesproken wordt. Er was een uitgebreid track met cloud gerelateerde sessies met daarin best-practice sessies van early-adopters, cloud providers die hun PaaS oplossing demonstreren en Java EE gerelateerde sessies over hoe met de cloud om te gaan. Inmiddels is het ook duidelijk dat de cloud niet alleen bedoeld is voor partijen die met een extreme hoeveelheid gebruikers of data om moeten gaan. Uiteraard is schaalbaarheid (en dan vooral elastisch schalen) een belangrijk onderwerp, maar daarnaast is voor veel partijen juist het gemak van deployment en onderhoud de reden om applicaties in de cloud te gaan draaien. Met

PaaS verdwijnt de applicatie server in de onderliggende infrastructuur en dit stelt dus belangrijke eisen aan de manier waarop we applicaties ontwikkelen en deploybaar maken. De ease-of-development slagen en het terugbrengen naar een POJO gebaseerd programmeermodel van veel Java API's en frameworks en de DevOps beweging (het laten samenkomen van development en operations) leveren daar een belangrijke bijdrage aan.

Modularity

Waar voorgaande jaren OSGi een verboden technologie leek te zijn op JavaOne was er dit jaar een goede vertegenwoordiging. Naast pure OSGi onderwerpen waren er ook nog een aantal sessies over modulariteit in het algemeen. Iedereen lijkt het er over eens te zijn dat modulariteit onmisbaar is, zeker als applicaties in de cloud gaan draaien. Over de juiste modulariteit oplossing is echter nog enige onenigheid. Alle ogen zijn hierbij gericht op project Jigsaw dat onderdeel gaat worden van Java 8.

Jigsaw gaat modulariteit onderdeel van de taal maken. Jigsaw is een aantal jaar geleden gestart om de JDK zelf modulair te maken. De JDK is door het toevoegen van steeds meer features en APIs in de afgelopen tien jaar enorm gegroeid en heeft inmiddels een flinke omvang. Van sommige APIs is de vraag ook hoe zinvol ze nog zijn omdat ze in moderne applicaties vrijwel niet meer gebruikt worden. Het meest voor de hand liggende voorbeeld hiervan is Corba ondersteuning. Tegelijkertijd is het onmogelijk om deze APIs zomaar weg te gooien, omdat dit de backwards compatibility zou breken en de paar applicaties die er nog wel van afhankelijk zijn zullen voor eeuwig worden veroordeeld tot een oude versie van Java. De enige oplossing voor dit probleem is om de JDK modulair te maken, zodat een eindgebruiker alleen die onderdelen van de JDK hoeft te downloaden die nodig zijn om een applicatie op te starten. Op JDK niveau is OSGi echter geen oplossing aangezien OSGi een niveau boven de JDK staat. De huidige plannen voor Jigsaw gaan echter veel verder dan het opsplitsen van de JDK. De modulariteit ondersteuning wordt ook beschikbaar gemaakt voor Java ontwikkelaars om eigen applicaties modulair te maken. En dat botst met OSGi. Jigsaw lost hetzelfde probleem op als OSGi op het "bundle" niveau van OSGi. Waar OSGi daar een extern framework en een extra metadata formaat voor nodig heeft, krijgt Jigsaw ondersteuning vanuit de Virtual Machine en de taal. Er is echter nogal wat discussie gaande of de manier waarop Jigsaw dit aanpakt wel de juiste is. Op dit moment lijkt het erop dat Jigsaw en OSGi twee aparte oplossingen worden, maar dat samenwerking tussen de twee wel mogelijk wordt. Over wat dit voor OSGi gaat betekenen zijn de meningen ook sterk verdeeld. In ieder geval gaat Jigsaw geen oplossing bieden op het niveau van "services" in OSGi. Dit is wellicht het onderdeel van OSGi dat het meeste voordeel biedt voor het echt modulair maken van applicaties. Een combinatie van module scheiding op basis van taalondersteuning met daarop OSGi services zou weleens een hele interessante invalshoek kunnen zijn. Het laatste woord is hier nog zeker niet over gesproken.

Java SE 8

Door diverse redenen is de release van Java SE 7 enorm vertraagd geraakt. Nadat al meer dan vijf jaar waren verstreken sinds versie 6 werden op JavaOne 2010 twee opties voorgesteld om hier mee om te gaan. Java 7 uitbrengen met een beperkter aantal vernieuwingen dan

oorspronkelijk gepland en de grote vernieuwingen in Java 8 onderbrengen, of de release datum verder opschuiven. Omdat een aantal onderdelen van Java 7 al klaar waren werd er gekozen voor de eerste optie. Java 7 is inmiddels beschikbaar (sinds juli van dit jaar) en tijdens JavaOne is ook de officiële Mac OS X versie voor eind dit jaar als beta aangekondigd, gevolgd door een GA (General Availability) release begin volgend jaar. Overigens is de TCK - de Technology Compatibility Kit waarmee wordt bepaald of een JVM inderdaad SE 7 gecertificeerd kan worden - nog niet gereed; Oracle geeft als reden een gebrek aan advocaten aan(!) en noemt het laatste kwartaal van 2011 als doel voor de TCK.

Omdat Java SE 8 een aantal echt grote vernieuwingen gaat brengen die toch wel wat tijd vergen en bovendien grote ondernemingen bij Oracle en bij het OpenJDK project aan hebben gegeven dat een periode van een jaar tussen twee major Java releases toch echt te kort is, gaat die release nog iets worden uitgesteld, namelijk tot de zomer van 2013. De belangrijkste onderwerpen in SE 8 blijven modulariteit (project Jigsaw) en project Lambda (voor closures en bulk parallel operations in Java collections zoals filter/map/reduce).

Door het verschuiven van de releasedatum is er ruimte gekomen om in Java 8 extra faciliteiten op te nemen, zoals de nieuwe Date Time API (JSR 310, ook wel bekend als JodaTime), opnieuw wat kleinere verbeteringen in de taal, Security updates, vernieuwde Networking, Type Annotations, een vernieuwde JavaScript engine (project Nashorn) en ondersteuning voor moderne device-functies als multi-touch user interface, kompas, versnellingsmeter, camera sensoren en locatie-bepaling. Een belangrijk doel met Java SE 8 voor Oracle is de verbeterde interactiviteit tussen verschillende talen op de JVM. De nieuwe JavaScript engine moet ondermeer een goed voorbeeld geven van deze interactiviteit die vervolgens ook met andere JVM-talen - Groovy en Scala zijn de twee meest prominente - kan worden toegepast. Het lijkt waarschijnlijk dat Project Avatar - zie elders in dit artikel - ook functionaliteit rondom HTML 5 en WebSockets zal inbrengen in Java 8.

Theme	Description/Content
Project Jigsaw EXPANDED	• Module system for Java applications and the Java platform
Project Lambda EXPANDED	• Closures and related features in the Java language (JSR 335) • Bulk parallel operations in Java collections APIs (filter/map/reduce)
Oracle JVM Convergence	• Complete migration of performance and serviceability features from JRockit, including Mission Control and the Flight Recorder
JavaFX 3.0	• Next generation Java client
JavaScript NEW	• Next-gen JavaScript-on-JVM engine (Project Nashorn) • JavaScript/Java interoperability on JVM
Device Support NEW	• Multi-Touch (JavaFX), Camera, Location, Compass and Accelerometer
Developer Productivity	• Annotations on types (JSR 308), Minor language enhancements
API and Other Updates	• Enhancements to Security, Date/Time, (JSR 310) Networking, Internationalization, Accessibility, Packaging/Installation
Open Source	• Open development in OpenJDK, open source additional closed components

Figuur: thema's en specificaties voor Java 8 SE

Met het OpenJDK project zijn onderdelen hiervan op korte termijn al zelf uit te proberen als de eerste Early Access downloads beschikbaar komen. Dit is opnieuw een teken van de door de community afgedwongen veranderingen. De volledige ontwikkeling van Java vindt hiermee transparant en openlijk plaats. Ondenkbaar voor de gewone productontwikkeling binnen Oracle.

Oracle hoopt vrij kort na Java SE 8 de JavaFX 3.0 API ook onderdeel te maken van de SE specificatie, ter opvolging (op termijn) van de huidige Swing API. Het toevoegen van JavaFX - zie verderop in dit artikel over de vernieuwingen op dit gebied - is redelijk controversieel. Tot nu toe is FX volledig buiten de JCP en OpenJDK om door Oracle zelf ontwikkeld. Op dit moment maakt het dus geen onderdeel uit van de Java standaard. Omdat het de bedoeling is dat FX op termijn Swing moet doen vergeten als de standaard manier om UI op het Java platform te doen is het echter wel zaak om het onderdeel te laten uitmaken van het Java SE platform. Hiervoor zal het eerst door de JCP moeten en dat is niet zomaar een formaliteit. Oracle is namelijk wel de steward van Java, maar heeft het wat dit betreft duidelijk niet alleen voor het zeggen binnen de JCP. Dit is een ontwikkeling die potentieel voor een hoop vuurwerk zou kunnen gaan zorgen in de komende jaren, maar tegelijkertijd ook het bewijs van een gezond functionerende JCP kan leveren.

Er zijn zelfs al plannen voor Java 9. Een interessant thema hier is een zelf tunende Virtual Machine om de afhankelijkheid van de (soms ondoorzichtige) command line parameters te verkleinen. Andere thema's die voorzichtig genoemd worden - naast uiteraard de verdergaande modernisering en verbetering van Java als programmeertaal - zijn bijvoorbeeld meer (en zelfs native) integratie van diverse talen op de JVM en betere integratie met native (O/S) libraries en voorzieningen. In lijn met de NoSQL beweging zijn voorzieningen gepland voor 'big data' -

grote, in-memory (of grid) data collecties - die de huidige Java geheugenbeperking van 2 Gigabyte voor arrays moeten oprekken. Ook zal Java 9 volgens de huidige plannen een nieuwe meta-object beschrijving gaan bieden die eenvoudiger object-uitwisseling tussen talen mogelijk moet maken. Meer ondersteuning voor geavanceerde concurrency op de multi-core platformen van heden en vooral toekomst is ook voorzien. Net als reification - een voorziening om de type informatie van objecten ook op runtime beschikbaar te hebben en door velen gezien als een hoogstnoodzakelijke correctie op de oorspronkelijke implementatie van Generics in Java SE 5.

Multi-tenancy staat ook op de lijst voor Java 9 - gericht op cloud-omgevingen en bedoeld om meerdere applicaties binnen dezelfde JVM gelijktijdig te laten functioneren. Hiermee verwant is ook de genoemde ondersteuning voor resource management voor cloud applicaties. Tegelijkertijd is ditzelfde thema ook op de kaart gezet voor de volgende versie van Java EE. Hoe dit exact terecht zal komen in welke distributie van Java is dus nog even afwachten.

Continuations zijn een geavanceerde programmeertaal voorziening - gebaseerd op lambda expressions - die een haast asynchrone aanroep van locale methodes ondersteunt waarbij een aangeroepen methode geen waarde direct teruggeeft maar van de aanroeper een object krijgt waarop de resultaten geplaatst kunnen worden. Voor Java 9 wordt overwogen ondersteuning van continuations onderdeel te maken van de taal.

Het is tevens de bedoeling dat Java SE en ME weer samenkomen - met een beperkte (core) footprint die op alle platformen beschikbaar kan zijn. Het weer overal - van Card tot EE en Cloud - kunnen toepassen van dezelfde Java taal is een nadrukkelijk streven voor de nabije toekomst.

De samenwerking in het OpenJDK project lijkt ook erg positief te zijn. Naast ondermeer Oracle, RedHat, IBM, SAP en Apple hebben ook Azul en Twitter(!) zich recent aangesloten. Het community proces lijkt ook hier dus weer goed te bewegen.

Het lijkt alsof Groovy van de JVM scripting-talen de meeste toepassing heeft gevonden - bijvoorbeeld gelet op de relatieve verhoudingen in het aantal vacatures:



De heel goede integratie van Groovy met Java speelt zeker een rol bij de veelgemaakte keuze om Groovy in Java en JEE applicaties in te zetten voor specifieke taken.



Om te bewijzen hoe goed de integratie tussen Java en dynamische scripting talen op de JVM wel niet kan zijn wordt enerzijds gewerkt in Java 8 SE aan nieuwe features op dat gebied en anderzijds aan een nieuwe JavaScript engine die van die integratie gaat profiteren. Project Nashorn moet deze JavaScript Virtual Machine gaan opleveren, de opvolger van Rhino. De nieuwe JavaScript engine moet zeer veel sneller zijn dan Rhino, vergelijkbaar met V8 (Google Chrome) en Mozilla *Monkey.

Java EE 7

Java EE 6 is ondertussen bijna twee jaar oud. Inmiddels wordt Java EE door zeven containers ondersteund en is daarmee op steeds meer plekken als mainstream technologie in de praktijk te gebruiken.

CAUCHO

FUJITSU



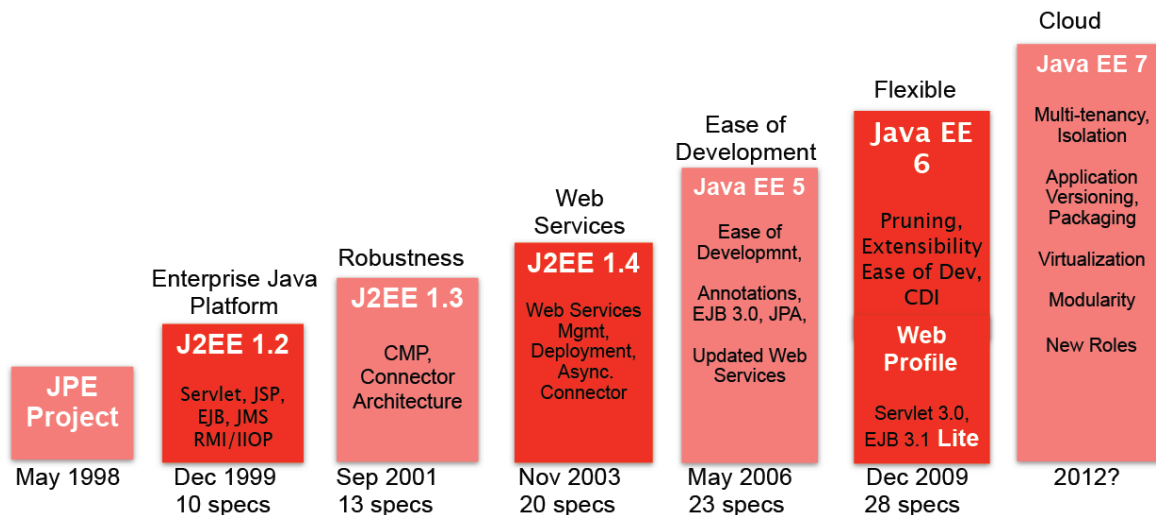
TmaxSoft

IBM

ORACLE

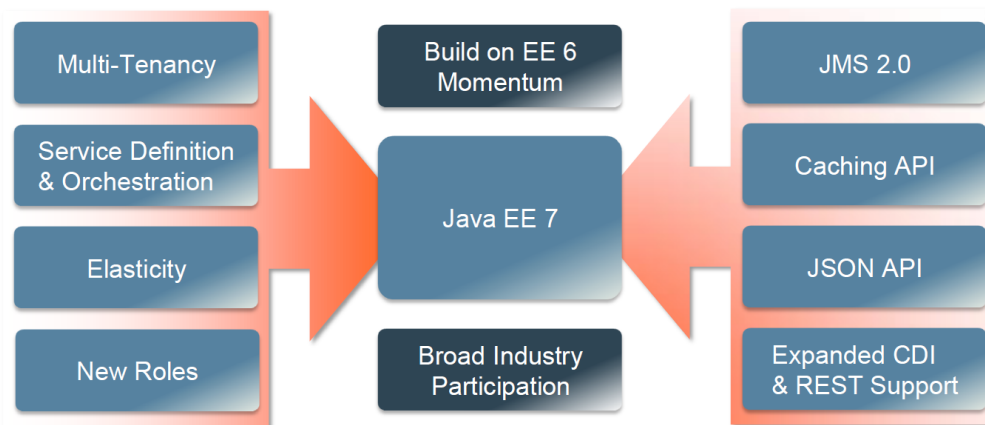
redhat

De meeste containers zijn ontzettend lichtgewicht en starten binnen enkele seconden, wat een enorme vooruitgang is ten opzichte van de traditionele application servers. Dat op zichzelf is ook weer een belangrijke stap richting de cloud. In de cloud wordt er vaak gewerkt met een model waarbij betaald wordt naarmate gebruik. Geheugen onnodig opslokken kost daarbij direct geld en een kleine geheugen footprint is daarmee belangrijk geworden. Daarnaast is de snelle opstarttijd belangrijk voor het goed kunnen op en neer schalen van een cluster in de cloud. Ondanks dat Java EE 6 hiermee voor de meeste ontwikkelaars nu pas in beeld komt, is de ontwikkeling van Java EE 7 inmiddels in volle gang.



Dat betekent dat JSRs en expert groups worden gevormd en dat nieuwe API's uitgedacht worden. De ene JSR gaat hierin een stuk harder dan de andere. Sommige specificaties zoals bijvoorbeeld JAX-RS 2.0 zijn al voor een groot deel klaar. De API toevoegingen zijn in een aantal container implementaties zelfs al aanwezig, al kunnen deze uiteraard nog veranderen. Dit komt omdat bijvoorbeeld JAX-RS 2.0 de gaten invult waar ontwikkelaars nu in de praktijk tegenaan lopen. Voor andere specificaties geldt dat er pas in een verkennende fase over nieuwe ideeën wordt nagedacht. Een voorbeeld hiervan is multi-tenancy support in JPA. Hier is ook absoluut community input nodig om dit op de juiste manier te vormen. Dit was ook een duidelijke oproep op JavaOne. De meeste JSRs zijn open en input leveren kan zo eenvoudig zijn als deelnemen aan een mailinglist. Een ander belangrijk statement over Java EE 7 is dat het niet *alleen* maar over de cloud gaat. Ondanks dat cloud ondersteuning een belangrijk thema is voor Java EE 7 wordt uiteraard ook op alle andere vlakken het platform verbeterd. Denk hierbij aan CDI 1.1, JMS 2.0 en EJB 3.2. Ook een Java API voor het manipuleren van JSON staat op het programma.

Java EE 7 – Design Objectives



Als het wel over cloud gaat zijn een aantal onderwerpen belangrijk voor Java EE 7:

- Multi-tenancy
- Cache/Datagrid APIs
- NoSQL
- Deployment

Een aantal dingen die nu al duidelijk zijn is dat er een API komt als abstractie op de verschillende cache en datagrid oplossingen die in cloud oplossingen veel gebruikt worden. Denk hierbij aan een abstractie zoals JDBC/JPA dat biedt voor relationele databases. Ook wordt er hard nagedacht over ondersteuning voor NoSQL. Dit is bijvoorbeeld een van de punten waar de JPA export group aan werkt. Het lastige hieraan is dat er nog geen duidelijk model is waar alle datastores aan voldoen, waardoor het ook heel lastig is om tot een bruikbare standaard te komen. Multi-tenancy is een nog veel complexer onderwerp. Multi-tenancy kan namelijk op hele verschillende niveaus worden opgelost. Multi-tenancy kan betekenen dat alleen infrastructuur gedeeld wordt, of misschien ook de application server, tot aan een model waarbij er alleen in de data onderscheid wordt gemaakt tussen tenants. Dit laatste is bijvoorbeeld een veel gebruikte oplossing voor SaaS. Het oplossen van dit probleem moet dus op diverse niveau's in het Java platform gebeuren.

Een idee van hoe multi-tenancy bijvoorbeeld in JPA zou kunnen worden uitgewerkt:

EMPLOYEE table

EMP_ID	VERSION	F_NAME	L_NAME	GENDER	DEPT_ID
1	1	John	Doe	M	1
2	3	Jane	Doe	F	2

```
@Entity
```

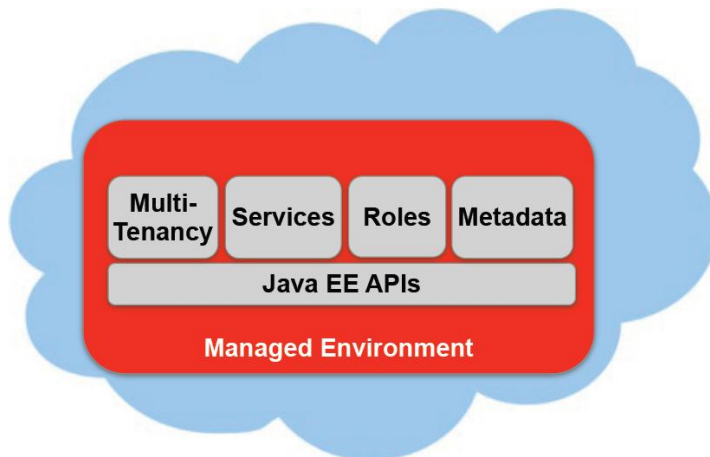
```
@Table(name="EMPLOYEE")
```

```
@MultiTenant(SINGLE_TABLE)
```

```
@TenantDiscriminator(name="department", columnName="DEPT_ID")
```

```
public class Employee {
```

Op beschouwend niveau is het wel belangrijk om vast te stellen dat Oracle voornemens is om Java EE het default programmeermodel van de cloud te laten zijn. De diverse cloud vendors bieden op dit moment een zeer diverse ondersteuning van subsets van het Java platform aan. De ene vendor gaat hier heel anders mee om dan de andere. Als Java EE geschikt gemaakt wordt cloud development platform, dan is daarmee een heel belangrijke stap gezet naar standaardisatie van een uniform, vendor breed cloud ontwikkelplatform en kan cloud worden toegevoegd aan het paradigma van 'write once, run anywhere'.



De inhoud van JEE7 is in elk geval: JMS 2.0, EJB 3.2, Servlet 3.1, JPA 2.1, JAX-RS 2.0, CDI 1.1, Bean Validation 1.1, JSF 2.2, EL 3.0 als herziene specificaties en als nieuwe JSRs: JCache 1.0 – Java Temporary Caching API, Concurrency Utilities 1.0, Java State Management 1.0, Java Identity API 1.0, Batch Processing 1.0 en JSON 1.0.

Extensions

Java EE 7 staat op de planning voor eind 2012. Dat lijkt nog een eind weg en de vraag is dan ook of dit niet te laat is. Dit is om twee redenen niet het geval. De eerste reden is dat Java EE 6 veel beter "extensible" is dan voorgaande versies. Via CDI kunnen op een standaard manier

uitbreidingen worden gedaan die tot diep in het platform kunnen inhaken. Met behulp van zulke uitbreidingen kan er innovatie worden gedaan bovenop Java EE 6, zonder dat hiermee het platform eigenlijk wordt weggegooid. In eerdere versies was het platform nauwelijks extensibel waardoor frameworks zoals Seam en Spring met een eigen programmeermodel moesten komen. Standaard extensies kunnen worden geïnnoveerd en ontwikkeld in de community en als uitbreiding op applicatie servers, en kunnen vervolgens in een volgende versie van de specificatie worden gestandaardiseerd. De lock-in op niet-standaard oplossingen wordt hiermee veel kleiner en het bestaansrecht van rebel-frameworks die een complete alternatieve programmeerstack bieden neemt hiermee af.



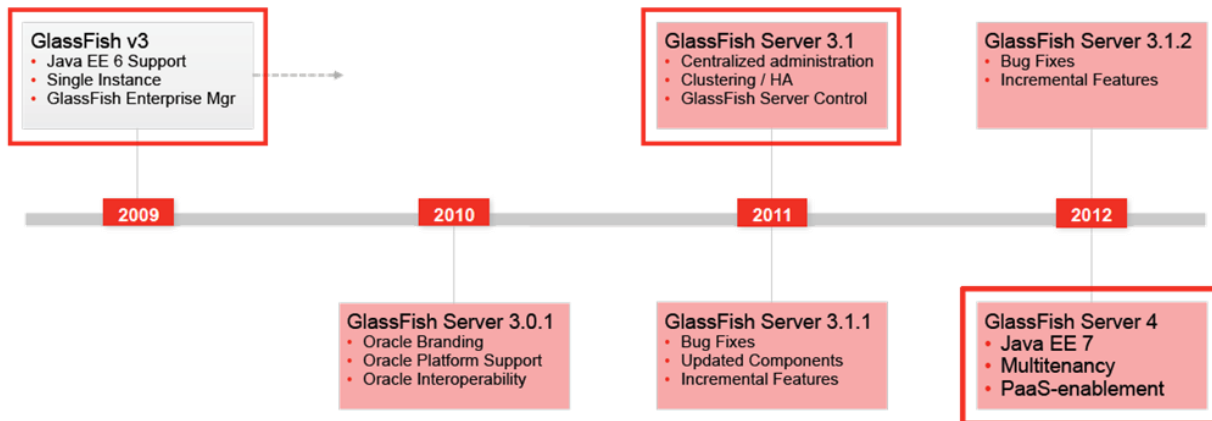
Glassfish

De referentie-implementatie voor JEE 7 (net als voor 5 en 6) is Glassfish - dit keer release 4.0. Glassfish is door Sun ontwikkeld en door Oracle omarmd. De ontwikkeling van Glassfish gaat volle kracht vooruit, naast en veelal hand in hand met de andere JEE applicatie server: WebLogic Server. Deze laatste wordt door Oracle met name gepositioneerd als de enterprise container en de hoeksteen voor Fusion Middleware en Exalogic. Voor 'afdelingsniveau' JEE applicaties is Glassfish ook uitstekend bruikbaar en de dit voorjaar uitgebrachte versie met clustering-support suggereert ook wel wat grotere ambities. De teams achter WebLogic en Glassfish wisselen code uit - bijvoorbeeld de JSF 2.0 implementatie op WebLogic en op Glassfish is dezelfde code - en wederzijds worden verbeteringen geleend.

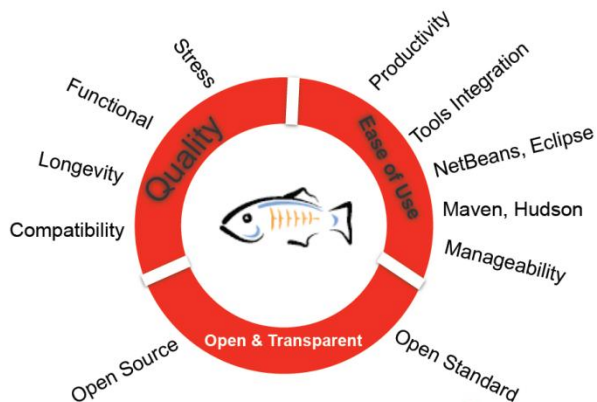
De eerste very early access builds voor Glassfish 4.0 zijn al beschikbaar voor download - met dus ook de eerste voorbeelden van wat JEE 7 aan nieuwe APIs en specificaties zal gaan herbergen.



Naast deze nieuwe 4.0 branch gaat ook de ontwikkeling van Release 3 verder:



De belangrijkste aandachtsgebieden bij de verdere ontwikkeling van Glassfish:



Spring vs Java EE

Mede ook in het kader van die afnemende relevantie van alternatieve programmeerstacks was het interessant om te zien dat er dit jaar voor het eerst op een JavaOne meerdere sessies waren die Spring en Java EE tegen elkaar zetten. Een van de sessies werd gepresenteerd door Bert Ertman en Paul Bakker en wordt herhaald op J-Fall en Devvxx. Naast het feit dat dit de eerste JavaOne was waarbij Java EE zo nadrukkelijk als alternatief voor Spring werd neergezet, was het nog veel interessanter om te zien dat hier zeer veel belangstelling voor was. Alle sessies gerelateerd aan dit onderwerp zaten helemaal vol. Java EE is duidelijk terug op de kaart en is in ieder geval onder JavaOne bezoekers uitermate populair.

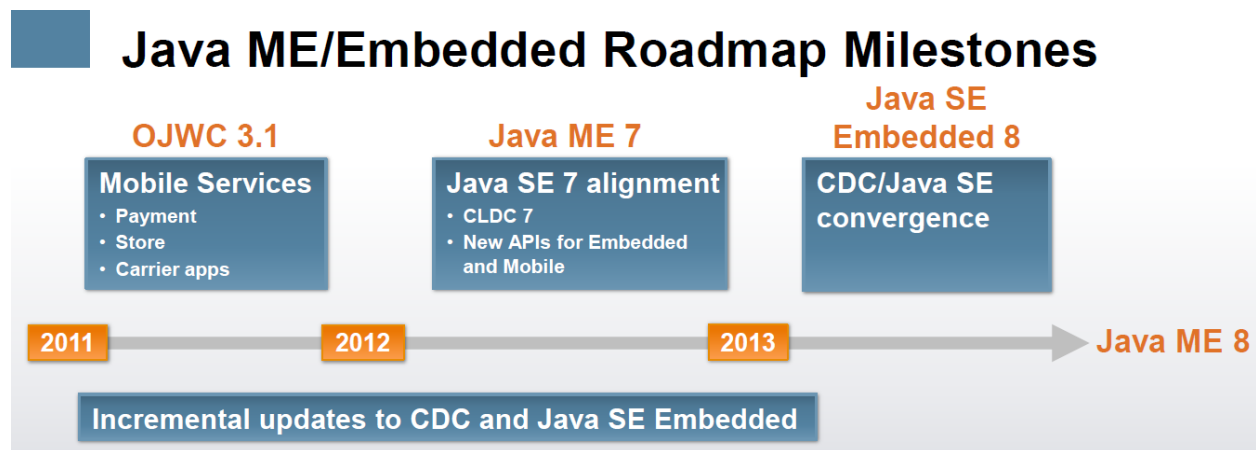
Java in het klein - ME en Card

Java draait niet alleen op hele grote machines en enterprise-omgevingen maar juist op veel meer miniatuur-omgevingen. Bijvoorbeeld op een paar miljard SimCards en embedded devices- in de vorm van JavaCard en de Java Micro Edition. Met de toenemende verspreiding van dit

soort devices en een toenemende connectiviteit - met Near Field Communication, BlueTooth en via telefoon-connectiviteit - lijken de mogelijkheden voor aanbieders - en dus Java ontwikkelaars - om extra waarde toe te voegen met behulp van het Java platform zeer aanwezig.

De roadmap op dit onderdeel van het Java platform heeft enkele grote doelstellingen:

- zoveel mogelijk aansluiten bij Java SE zodat er weer sprake is van één platform en taal (write once, run anywhere)
- gebruikmaken van de modulariteit die in Java 8 SE in de JVM wordt geïntroduceerd om weliswaar de standaardimplementatie van de JVM te kunnen benutten, maar daar een relevante sub-sectie van te gebruiken
- Java-ontwikkelaars een standaard API bieden om services van telefoon-dienst-aanbieders (zoals locatie, SMS, betalingen) en telefoon-functies voor bijvoorbeeld near field communication



REST

Ook in recente edities van JavaOne was REST(ful) een prominent aanwezig onderwerp. Om verschillende redenen. REST wordt ervaren als eenvoudiger dan SOAP/WS* WebServices en daarom omarmd. Daarnaast kan REST in combinatie met JSON als data formaat eenvoudig worden geconsumeerd op mobiele devices en in JavaScript web clients. Deze JavaOne conferentie liet zien dat er nog een reden ontstaat voor de massale aandacht voor REST: de cloud! Als deployment van Java applicaties op cloud infrastructuur meer en meer gebruikelijk wordt, zal ook de administratie van deze JVMs in de cloud meer en meer aandacht vragen. Voor veel beheersactiviteiten bieden de bekende applicatie servers naast web gebaseerde consoles veelal ook programmatische middelen, gebaseerd bijvoorbeeld op MBeans en JMX, op EJBs of op proprietary protocollen zoals bijvoorbeeld t3 bij WebLogic. Deze programmatische mechanismen werken niet over HTTP naar de Cloud. RESTful APIs zijn dan een voordehandliggend alternatief. En inderdaad is dit het mechanisme dat wijd en breed wordt omarmd voor de volgende generatie applicatie servers.

REST op zich is niet nieuw en de JAX-RS 1.1 API is onderdeel van Java EE 6. In JSR 339 wordt JAX-RS 2.0 ontwikkeld, bedoeld als onderdeel van JEE 7. De betreffende Expert Group omvat naast Oracle ondermeer Ebay, RedHat en Talend vertegenwoordiging. Met JAX-RS 2.0. gaat de volwassenwording van REST verder en wordt de toepassing ook verbreed. Naast voornamelijk interactiekanaal tussen client en server wordt REST nadrukkelijker ingezet als server-naar-server integratie mechanisme, een low-overhead alternatief voor hoog frequente (SOAP) webservice interactie. Ook verschillende data grids, waaronder Oracle Coherence, bieden RESTful APIs om data op te vragen en weg te schrijven.

Een aantal kernelementen in de JAX-RS 2.0 specificatie: betere client libraries voor consumptie van RESTful services, gestandaardiseerde Filters en Handlers (voor standaard interceptie en bewerking van binnenkomende en uitgaande berichten), validatie van parameters en resources met behulp van Bean Validation, asynchrone interactie (op basis van de asynchrone afhandeling van Servlet 3.0) en ondersteuning voor links (verwijzingen naar andere RESTful resources in een RESTful response).

JavaFX

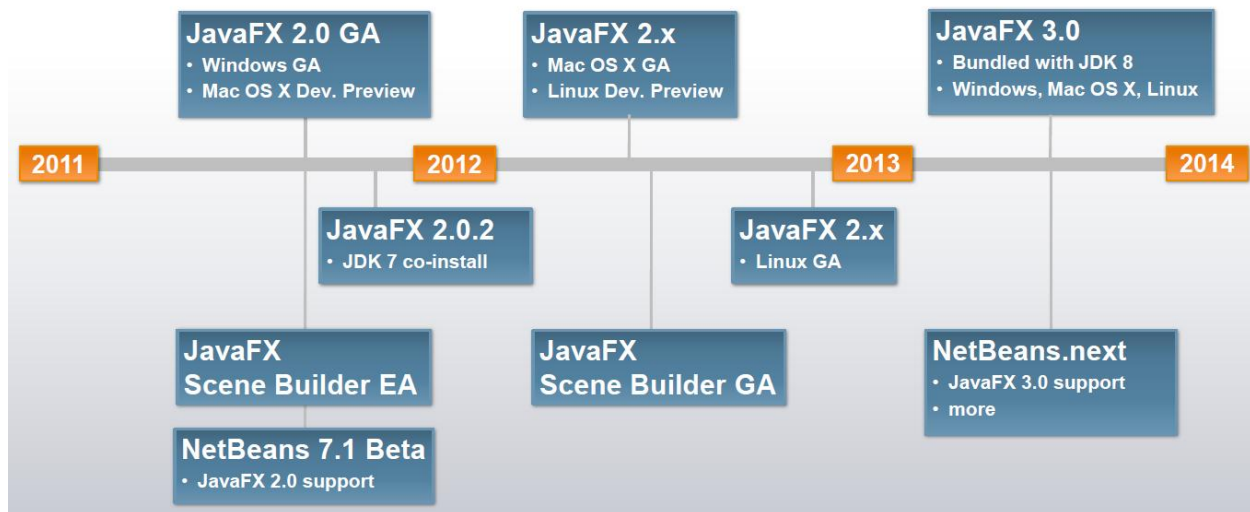
JavaFX was op JavaOne prominent aanwezig. Hierbij was het vooral duidelijk dat JavaFX niet langer een speelgoed technologie is, maar strategisch belangrijk voor Oracle. Het was dan ook niet verwonderlijk dat er een aantal belangrijke aankondigingen zijn gedaan:

- JavaFX wordt in de loop van volgend jaar volledig open source, op een vergelijkbare manier als OpenJDK.
- Er komt een specificatie (JSR) voor JavaFX
- De Mac versie van JavaFX wordt dit jaar nog uitgebracht (de beta is reeds beschikbaar)
- De Linux versie volgt begin 2012

Tijdens de keynotes werden er een aantal zeer indrukwekkende demo's getoond van JavaFX applicaties. Vooral de 3d scenes met geïmporteerde modellen geven aan hoe krachtig de technologie op grafisch gebied is. Ook waren er een aantal demo's waarop JavaFX applicaties op tablets draaiden. Windows en Linux (als alternatieve benaming voor Android omdat dit wat gevoelig schijnt te liggen) tablets lagen voor de hand, maar de demo waarbij een JavaFX applicatie op een iPad draaide werd erg gewaardeerd door de bezoekers. iOS ondersteuning staat echter nog niet op de roadmap, de community wordt gevraagd hoe belangrijk dit is. Geef hier ook vooral je mening over.

Zoals vorig jaar al aangekondigd werd is JavaFX nu een echte Java technologie. JavaFX Script bestaat niet meer en een JavaFX applicatie wordt nu gewoon in Java geschreven. Daar zijn drie APIs voor beschikbaar; een traditionele API die lijkt op Swing, een op het builder pattern gebaseerde API en een op XML gebaseerde variant om markup mee te doen. Op deze manier kunnen ontwikkelaars met verschillende achtergrond goed uit de voeten met de API. Omdat er geen aparte taal meer is kunnen alle IDEs ook direct gebruikt worden voor het ontwikkelen van JavaFX applicaties.

JavaFX Roadmap



Swing is nadrukkelijk dood verklaard. De politiek correcte versie daarvan is dat er momenteel vrijwel alleen nog maar bugs worden opgelost in Swing en er weinig nieuwe ontwikkeling meer plaats zal vinden. Een goed verstaander weet echter wat dit betekent. Gelukkig is de integratie tussen Swing en JavaFX nu goed mogelijk en komen er duidelijke migratiepaden. JavaFX is daarmee ook niet meer puur gepositioneerd als RIA concurrent voor Silverlight en Flex. JavaFX moet worden gezien als de standaard Java GUI technologie van de toekomst. De manier waarop de GUI wordt aangeboden - desktop of web - wordt bepaald door de onderliggende renderkit. Voor het web moeten we het vooralsnog doen met Java Web Start en Applets als onderliggende mechanisme. Voor de toekomst staat rendering naar HTML 5 op het programma.

Nieuwe JMS 2.0 specificatie - in wording

Vorig jaar schreven we over een community bijeenkomst tijdens JavaOne 2010 waar 's avond om tien uur in een zogenaamde Birds of a Feather sessie door pakweg 40 man over de toekomst van JMS werd gefilosofeerd. Op dat moment was Java EE 7 nog niet in de steigers gezet en was er ook nog geen duidelijkheid over een JMS 2.0 release. In deze BoF werden ideeën uitgewisseld over hoe JMS verbeterd zou kunnen worden en wat tekortkomingen waren. Nu, een jaar later, is Java EE 7 volop in beweging, is er inderdaad een JMS 2.0 specificatie (JSR 343) in de maak door een expert group waarin ondermeer Tibco, RedHat, VMWare, Caucho, IBM en Oracle zijn vertegenwoordigd. Tijdens JavaOne2011 werd er opnieuw over gesproken in een BoF sessie.

Doelstelling van deze JSR zijn de opschoning van de aloude JMS 1.1 interfaces (stammend uit 2003), het moderniseren a la Java EE 5 en 6 met annotaties, unchecked exceptions en injectie, het ondersteunen van features die in veel messaging-producten beschikbaar zijn - zoals

minimum delivery delay, message delivery bevestigingsberichten, delivery count en batch delivery - en het oplossen van enkele MDB onvolkomenheden.

Een consequentie van de voorgestelde wijzigingen zou ondermeer zijn dat deze code:

```
public void sendMessage (String payload) {
try {
    Connection conn = null;
    con = cf.createConnection();
    Session sess =conn.createSession(false,Session.AUTO_ACKNOWLEDGE);
    MessageProducer producer = sess.createProducer(dest);
    TextMessage textMessage = sess.createTextMessage(payload);
    messageProducer.send(textMessage);
} catch (JMSEException e) {
    // do something
} finally {
    connection.close();
}
}
```

kan worden herschreven tot:

```
@Inject @JMSConnection(lookup="jms/connFactory")
@JMSDestination(lookup="jms/inboundQueue") MessageProducer producer;
public void sendMessageNew(String payload){
    producer.send(payload);
}
```

Project Avatar - HTML 5 en WebSockets

In de strategie keynote kondigde Oracle ook project Avatar aan. De evolutie van HTML 5 vormt de aanleiding voor dit project. Een op het eerste oog wat onduidelijk project voor een heel duidelijke ontwikkeling. HTML 5 wordt steeds meer gezien als ondermeer de vervanger van Flash voor rijke visualisaties in browsers en op mobiele devices. Vaak geassocieerd met HTML 5 is de opkomst van het WebSockets protocol als lichtgewicht bi-directioneel communicatieprotocol dat echte push mogelijk maakt - zowel tussen web server en web client of mobiele client maar ook tussen servers onderling. Met project Avatar wil Oracle bewerkstelligen dat in alle onderdelen van het Java platform - SE, EE, ME en JavaFX - ondersteuning wordt gerealiseerd voor HTML 5 en WebSockets, en dat die ondersteuning bovendien consistent is over het platform heen.

De scope van project Avatar is 'een complete oplossing voor dynamische rijke client applicaties' en noemt met nadruk de Java EE (Cloud) als server en HTML 5 browser applicaties, hybrides van HTML 5 en Java in browser of op mobiel device en (desktop) applicaties als client met communicatie op basis van JSON via WebSockets. De bi-directionele communicatie moet binnen Avatar worden benut om tot event-gebaseerde applicaties te kunnen komen - met

ondermeer notificaties bij data-wijzigingen. Avatar omvat ook een gestandaardiseerde aanpak van off-line faciliteiten voor web applicaties. Aangezien het hier met name om slide-ware ging en er geen concrete voorbeelden werden getoond valt het op dit moment nog te bezien hoe Avatar uiteindelijk van invloed zal zijn op de evolutie van het Java platform. De gedachte erachter is echter zeker van belang om Java als platform mee te laten evolueren met het programmeermodel van het web.



Successierechten - Oracle en de erfenis van Sun

Oracle als Steward van Java lijkt dus (in de herkansing) niet tegen te vallen. De verhoudingen binnen de Java community stabiliseren. Het aantal deelnemers aan JCP en OpenJDK neemt weer toe. De samenwerking tussen commerciële concurrenten als IBM, Apple, Oracle, SAP en RedHat/JBoss ten aanzien van het Java platform verloopt vruchtbaar - afgezien van de wat ongelukkige situatie met Google, de grote afwezige op deze conferentie.

Oracle heeft de boedel van Sun Microsystems ondertussen uitgebreid bestudeerd. Sommige onderdelen van die boedel zijn liefdevol in de armen gesloten - zoals JavaFX, NetBeans en Glassfish - en andere zijn gedecideerd aan de rand van de straat gezet - bijvoorbeeld JCAPS, OpenESB, project Kenai en jMaki. De geluiden vanuit de voormalige Sun gelederen zijn grotendeels positief. Na jaren van beperkte armslag door de financiële situatie van Sun is er in de afgelopen anderhalf jaar voor de geselecteerde projecten aanzienlijk meer ruimte ontstaan voor investering, uitbreiding en door groei. De stagnatie die sinds 2008 was ontstaan in allerlei aspecten van Java - platform en community - lijkt doorbroken. En daar profiteert vrijwel iedereen van.

Neem bijvoorbeeld de Sun Hotspot JVM. Oracle had - via de BEA overname - al een JVM in bezit: JRockit. Deze JVM werd commercieel aangeboden voor veeleisende omgevingen waar vergaande beheer en fine tuning faciliteiten benodigd waren en ook de garbage collection voorspelbaar en beter instelbaar moest zijn. Oracle voelt er weinig voor om twee JVMs te gaan onderhouden en heeft er voor gekozen om JRockit en Hotspot te laten samengaan in een JVM -

met voordelen en sterke punten van beide gecombineerd. Deze convergentie vindt plaats in de OpenJDK en zal in een aantal stappen verlopen tot aan of zelfs na JDK 8. Allereerst zijn JCMD command line tool, JMX Agent en MBeans, JDP (Java Discovery Protocol) en Flight Recorder aan de beurt (al voor JDK 7). Er blijft een commerciële JRockit editie bestaan met enterprise support, maar zonder die support-optie is JRockit nu onder dezelfde licentie verkrijgbaar en dus gratis bruikbaar als Hotspot JVM.

Glassfish was ook zo'n product dat overlapt met een commercieel product van Oracle, namelijk WebLogic Server (ook van BEA overgenomen). Glassfish is de referentie implementatie van Java EE en daarmee een hoeksteen van de specificatie. Sun bood Glassfish zowel als open source applicatie server aan als ook met een commerciële, enterprise support licentie. De community was benieuwd naar de plannen van Oracle.

Ook ten aanzien van Glassfish geldt dat Oracle geen trek heeft in het onderhouden van twee volledige Java EE applicatie servers. Aan de andere kant wil het wel degelijk serieus geld verdienen met WebLogic Server terwijl het ook gecommitteerd is aan de Java EE specificatie en de Glassfish referentie implementatie. De oplossing is ook hier dat de teams voor Glassfish en WebLogic Server nauw zijn gaan samenwerken en dat over en weer features en libraries worden uitgewisseld en gedeeld. Zo is de JSF library in WebLogic nu dezelfde als in Glassfish en wordt er ook rondom de Metro WebServices stack code gedeeld. Op termijn zou het er op uit kunnen draaien dat WebLogic een extensie is van Glassfish - een commercieel aangeboden verrijking met geavanceerde features als Coherence WebCache, Clustering en Administration op basis van Glassfish als Java EE 7 fundament.

Naast een referentie implementatie is Glassfish ook een showcase van de nieuwe ideeën die er leven met betrekking tot het cloud geschikt maken van Java EE 7. Dit geeft input aan de JCP expert groups die er mee aan de slag gaan. Glassfish is daarmee een veelzijdig product geworden wat de status van speeltuin Java EE product al lang ontstegen is. Dit betekent in elk geval dat Glassfish ook onder de hoede van Oracle - of misschien nog wel sterker dan voorheen - als complete, ondersteunde en qua features per definitie vooroplopende Java EE container kan worden ingezet. De verspreiding van Glassfish zou hierdoor wel eens versterkt kunnen gaan doorzetten. Tijdens de Glassfish BOF werd een foto getoond van het team achter Glassfish. Een panoramashot met een kleine honderd mensen.

De geruchten in de wandelgangen willen overigens doen geloven dat Oracle plannen heeft om een community edition uit te brengen van het ADF ontwikkelframework. Deze zou Glassfish als (voornaamste) platform kennen. Deze variant van ADF is niet volledig - bijvoorbeeld ondersteuning voor desktop integration (Excel) en mobile ontbreekt waarschijnlijk - maar is wel uitgebreid, inclusief de ADF controller, data binding en business components naast de ADF Faces rich client components.

'Als we er vandaag over zouden moeten beslissen, zouden we er nooit aan beginnen' was het commentaar van enkele Java boba's binnen Oracle. 'Maar nu het ons in de schoot is gevallen zou het zonde zijn om er niet volop mee aan de slag te gaan.' Het gaat hier om JavaFX - de

door Sun in 2008 tijdens JavaOne gelanceerde technologie voor de user interface van morgen. Ondanks alle rumoer rondom en belangstelling voor JavaFX kwam de technologie niet echt van de grond. Vlak voor de overname van Sun noemde Larry Ellison in zijn keynote tijdens JavaOne JavaFX als een van de interessante pareltjes in de Sun collectie. En inderdaad, Oracle is stevig doorgegaan waar Sun gebleven was. JavaFX Script is losgelaten en er voor in de plaats gekomen zijn Java APIs. Dit is een stap op weg naar het opnemen van JavaFX in SE - waardoor JavaFX definitief als opvolger en vervangen van Swing en AWT kan worden omarmd. Niet alleen ten aanzien van desktop maar ook binnen Applets, met HTML 5 rendering in gewone browser based pagina's en wellicht toch ook weer op mobiele devices moet JavaFX de Java UI technologie-of-choice worden. Of het lukt met JavaFX hangt zeker niet alleen van Oracle af - maar ook dit stuk Sun erfgoed wordt in ere gehouden.

De overname van BEA leverde Oracle de Workshop plugin voor Eclipse op en met Sun kreeg het zelfs een volledige IDE in handen, te weten NetBeans. Wie zou denken dat gezien de rationalisaties op andere terreinen Oracle van drie IDEs - JDeveloper, Eclipse en NetBeans - toch wel heel snel zou willen terugsnijden naar een blijkt er naast te zitten. Alle drie de IDEs blijven ondersteund. NetBeans is en blijft het platform ter 'popularisering' van het Java platform in al zijn facetten. NetBeans wordt dus als eerste uitgerust met ondersteuning voor de nieuwste versies van Java SE, Java EE en JavaFX en wordt door Oracle ook neergezet als de IDE voor andere JVM talen, zoals JRuby, Groovy en Scala. Het is overigens wel een opdracht aan de community om die non-Java ondersteuning in te vullen in het NetBeans platform. NetBeans 7.1 kwam tijdens JavaOne uit, in Beta. Deze release biedt ondermeer ondersteuning voor JavaFX 2.0. Andere nieuwe features zijn CSS3 support, tools voor visual debugging van Swing en JavaFX user interfaces, in de IDE geïntegreerde ondersteuning voor Git, en nieuwe PHP debugging features. Naast NetBeans ontwikkelt Oracle ook het Oracle Enterprise Pack for Eclipse (OEPE) steeds verder - met ondermeer integratie met WebLogic Server (daar ligt ook het commerciële belang) en een steeds verdere ondersteuning van ADF ontwikkeling (wellicht vooruitlopend op een aanstaande community edition). De JDeveloper IDE is de derde in het rijtje; deze IDE blijft de strategische ontwikkelomgeving voor Oracle Fusion Middleware producten - van ADF en WebCenter tot aan SOA Suite, BPM Studio en Data Integrator.

Ondertussen op Oracle Open World

Tegelijkertijd met JavaOne werd een paar blokken verderop, in en rondom het Moscone conferentiecentrum, de Oracle Open World conferentie gehouden. Sinds de overname van Sun Microsystems door Oracle en de cruciale rol die Oracle speelt met betrekking tot Java zijn deze evenementen niet meer volledig los van elkaar te zien - en delen ze bijvoorbeeld de logistiek en het concert dat op woensdagavond door ondermeer Sting en Tom Petty & the Heartbreakers werd gegeven. De ontwikkelingen binnen 'de rest van Oracle' zijn in meer of mindere mate verbonden met die van de Java gemeenschap en het Java platform.

De grootste aankondiging van Oracle voorman Larry Ellison afgelopen week was de lancering van de Oracle Public Cloud - <http://cloud.oracle.com>. Na jarenlang door te hameren op 'grid computing' (de g in de 10g en 11g aanduiding van veel van Oracle's producten) en het belang

van private cloud, is Oracle nu toch door de bocht: voor kleinere ondernemingen is het niet realistisch om een eigen private cloud op te zetten met serieuze kans op benutting van elasticiteit en kwaliteitsverbetering en/of kostenverlaging door schaalvergroting. Dus komt ook Oracle met een Public Cloud aanbieding. Deze moet zich onderscheiden van andere clouds op het feit dat Oracle dezelfde technologystack - PaaS en SaaS - aanbiedt in zowel de public cloud als voor private cloud en andere on-premise toepassingen.

'In de Oracle cloud kan je precies dezelfde applicaties deployen als die je lokaal installeert' is de belofte. Zo kan je op de Oracle Java Cloud - een van de smaken cloud die wordt aangeboden - standaard JEE applicaties en ook ADF applicaties uitrollen. Althans, zo spreekt de marketing-machine. Tegelijk blijkt uit de technische specificaties dat toepassing van bijvoorbeeld JMS, EJB en JNDI enigszins beperkt is en ook zal de toegang tot het lokale filesysteem van de applicatie server wel niet volledig beschikbaar zijn. De Oracle Cloud gaat op 1 november live en kan gedurende 30 dagen via een trial gratis worden uitgeprobeerd. De cloud bevindt zich initieel in Austin, Texas en krijgt neven-vestigingen in Virginia, Schotland, Singapore, India en China. Klanten kunnen aangeven op welke locaties hun data en applicaties zich wel en niet mogen bevinden.

Een andere cloud gebaseerde dienst die Oracle lanceerde is Oracle Social Networking. Het betreft een soort combinatie van Twitter en Facebook, gericht op gesloten communities van bedrijven, hun medewerkers en eventueel hun externe contacten. Dit lijkt sterk op Yammer. Deze dienst is gebaseerd op de Oracle WebCenter producten.

Een unlikely suspect in de wereld NoSQL zou je kunnen zeggen. Als er een bedrijf is dat gegrondvest is op SQL dan is het wel Oracle. Toch was een van de grote aankondigingen tijdens Oracle Open World de lancering van de Oracle NoSQL database in combinatie met ondersteuning voor Hadoop en tools voor het laden van de uitkomsten van jobs uitgevoerd door een Hadoop netwerk in een Oracle database.

De NoSQL database is grotendeels een oude bekende: de Berkeley DB (BDB) database die Oracle in 2006 via de SleepyCat overname in huis haalde. Deze lichtgewicht database is niet relationeel opgezet maar georganiseerd via key-value pairs. Via Java APIs, een XQuery interface en sinds kort de SQLite API kan met BDB worden geïntegreerd. BDB ondersteunt atomic transacties, locking en replicatie en kan op een groot aantal platformen worden ingezet, met ondersteuning voor data-volumes tot 256 Tb. De meeste Linux distributies bevatten BDB.

Oracle biedt ook een appliance aan voor NoSQL: de BigData Appliance. Deze machine is geoptimaliseerd voor real-time verwerking van grote data volumes - bijvoorbeeld de 'digital exhaust': data die voortdurend wordt geproduceerd en die we in het recente verleden negeerden omdat we domweg niet de middelen hadden om er zinnige dingen mee te doen maar die nu misschien toch de moeite waarde lijken te worden; voorbeeld zijn ondermeer Twitter-feeds, blogs, click-en-navigatie-gedrag op websites, sensor-metingen, gedetailleerde track & trace data en netwerkstatistieken. Verwant aan de NoSQL productlijn is de Exalytics machine (met veel geheugen), de nieuwste telg uit de Exa-familie die ook Exadata (veel slimme

storage) en Exalogic (veel processor-kracht) omvat. Sleutelement in deze machines is het Infini-band netwerk - een lossless protocol dat zeer veel snellere datauitwisseling mogelijk maakt dan 'klassiek' ethernet. De betekenis van het netwerk, de opkomst van slimmere netwerk-protocollen en aandacht voor fysieke (co-)locatie van componenten lijkt in bredere zin toe te nemen. Bij verdere opschaling, distributie, grid-caching en cloud-benutting beginnen netwerksnelheden en fysieke afstand een merkbare invloed te krijgen op systeem-prestaties.

Grote nieuwe aankondigingen rondom de Oracle Fusion Middleware stack waren er eigenlijk niet. Eind 2012 wordt de 12c release verwacht van ondermeer WebLogic (dan eindelijk volledig EE6), ADF en JDeveloper, SOA Suite en Oracle Service Bus (dan grotendeels geïntegreerd, zowel runtime als design time) en BPM Suite.

Aandacht voor de business en de bedrijfsprocessen waren een kernthema op Oracle Open World. Dat vertaalt zich ondermeer naar real-time inzicht in de uitvoering van bedrijfsprocessen door middel van business dashboards en faciliteiten voor het op run-time aanpassen van het gedrag en het uiterlijk van applicaties en services (door Oracle aangeduid met design time @ run time). 'Empowering the Business User' was een veel gebezigd adagium. Een techniek die in het kader van de real-time, pro-actieve benadering van eindgebruikers vaak in presentaties langskwam was 'push' - veelal in combinatie met WebSockets en HTML 5.

Een ander veelbesproken thema was de toepassing van mobiele devices, zeker de iPhone en iPad naast meer Java-achtige Android apparaten, als uitsteeksels van 'de enterprise'. Voor dit soort toepassingen acht Oracle een compromis acceptabel: geen native apps die voor ieder device grotendeels opnieuw ontwikkeld zouden moeten worden, maar gebruikmaken van PhoneGap om een native aandoende app te presenteren op basis van HTML 5 die op een enkele generieke set broncode is gebaseerd en via extensies van PhoneGap wel met de device specifieke opties als camera, contacten, email, GPS en microfoon kan integreren. Opvallend is dat Oracle klaarblijkelijk al afspraken met Apple heeft gemaakt die het mogelijk maken op iOS een soort mini-JVM te draaien die Java code kan uitvoeren en ondermeer de interactie met file systeem, lokale database en RESTful web services mogelijk maakt. Daarnaast werkt Oracle met Apple vruchtbaar aan een manier om deze 'enterprise apps' buiten de AppStore te houden maar wel eenvoudig distribueerbaar te maken. Oracle ontwikkelt ADF Mobile op bovenstaande wijze en liet hier tijdens JavaOne de eerste demo's van zien.

Conclusie

Na jaren van zeer geringe evolutie en bijna stilstand, die ten opzichte van andere technologie als dynamic languages, .NET, iOS/Objective C feitelijk achteruitgang betekende, is het Java platform over de volle breedte weer in beweging gekomen, getuige onder meer de release van SE 7, (zelfs op Mac OS X) de plannen voor SE 8, de zeer concrete ontwikkelingen en ambities met EE 7 en het serieus doorstarten van JavaFX. Oracle lijkt de verantwoordelijkheden die horen bij het stewardship van Java inmiddels serieus te nemen (ook uit welbegrepen eigenbelang); er is weliswaar een 'rationalisatie' geweest waarin menig Sun project is gesneuveld - tegelijkertijd heeft Oracle de investeringen in andere projecten sterk vergroot ten opzichte van wat Sun deed en zich kon veroorloven in haar nadagen. De beloften uit 2010 over

de voortzetting van initiatieven en de concrete deliverables zijn grotendeels waargemaakt in het afgelopen kalenderjaar. Na enig aftasten en hier en daar wat knetteringen is een omvangrijke, breed gedragen en groeiende community van een aantal belangrijke vendors - met name IBM en daarnaast Twitter, Azul, SAP en RedHat/JBoss, VMware/SpringSource - waarin serieuze voortgang en noodzakelijke samenwerking ontstaan met Oracle (ook al zijn de controverses met ondermeer Google ongelukkig). De cijfers rondom Java blijven indrukwekkend: de aantallen devices waarop Java in een of andere vorm draait, de wereldwijde aantallen ontwikkelaars en leveranciers zijn enorm. Het momentum van Java bleef in het afgelopen jaar en blijft daarmee ook voor de toekomst bestaan. Iedereen die vol blijft houden dat Java ten dode is opgeschreven of zelfs al overleden is, is een populaire naprater van suggestieve blogs of wil geen oog hebben voor de daadwerkelijke feiten. Het is daarmee nog lang geen “..en ze leven nog lang en gelukkig”. Oracle zal er ook de komende jaren alles aan moeten blijven doen om de community aangehaakt te houden en de politiek van de JCP het hoofd te bieden. De concrete ontwikkelingen en zeker de plannen en ambities in de volledige breedte van het platform wekken het vertrouwen dat Java weer vooraan kan lopen, een moderne taal en platform is en niet de moderne legacy lijkt te worden die zich de afgelopen jaren begon af te tekenen. Om modern bezig te zijn, hippe dingen te doen, aantrekkelijke UIs te ontwikkelen, de nieuwste technologie te ontsluiten en te blijven profiteren van moderne inzichten in programmeertalen en software engineering kan je gewoon verder met het Java platform, al is dat niet in alle gevallen in de programmeertaal Java zelf. Door het aanhaken van een aantal populaire dynamische taalalternatieven op de JVM lijkt de wildgroei aan initiatieven die zich steeds verder verwijderden van Java en JVM enigszins afgeremd: aangesloten blijven op Java in de breedste zin van het woord lijkt weer een nuttig streven - in plaats van zo hard mogelijk tegen Java/JVM in te gaan.

Resources

Zelf iets zien van JavaOne 2011? Dat kan eenvoudig. Bijvoorbeeld alle keynotes zijn beschikbaar als videostream op <http://www.oracle.com/javaone/index.html>. Op deze pagina vind je ook een URL naar de JavaOne 2011 Content Catalog waar je PDFs met slides van vrijwel alle sessies kunt downloaden. Zo'n 40% van de sessie zal tevens beschikbaar worden gesteld als slidecast (met video van de demo's!) op Parleys.com. Direct na JavaOne zijn er een twintigtal sessies online gezet. Vervolgens zullen er iedere week drie nieuwe sessie worden vrijgegeven. Daarnaast is op een groot aantal blogs materiaal rondom JavaOne terug te vinden - van reviews en persoonlijke indrukken tot source code en presentatieslides.

JavaOne 2012 staat nu al in de agenda: van 30 september tot 4 oktober 2012 in San Francisco. Daaraan voorafgaand worden ook JavaOne conferenties gehouden in Brazilië (december 2011) en Tokyo (april 2012). Dichter bij huis is de J-Fall conferentie die de NLJUG op 2 november in Nijkerk organiseert overigens een prima eendaags alternatief, met diverse reprises van JavaOne en andere presentaties die daar zeker niet hadden misstaan. Datzelfde geldt voor de Devovx conferentie, 14-18 november in Antwerpen.