



Productiviteitsverhoging met **Application Platform as a Service**

Koen Vanderkimpfen
Bert Vanhalst

Sectie Onderzoek



Agenda

1. Inleiding
2. Benefits
3. Interne werking
4. Eigenschappen
5. Marktoverzicht
6. Impact op taken
7. Conclusies





Inleiding

Application PaaS

De Cloud komt naar de Software Ontwikkelaar

« Let developers just develop »
(Maak abstractie van onderliggende infrastructuur)

- **Snellere Time to Market**
- Automatisatie, Standaardisatie
- Efficiënter gebruik van middelen



Intro – Benefits – Interne werking –  – Eigenschappen – Markt – Impact – Conclusies

Application PaaS



- Opgepasst voor de public cloud...



De Cloud « Stack »

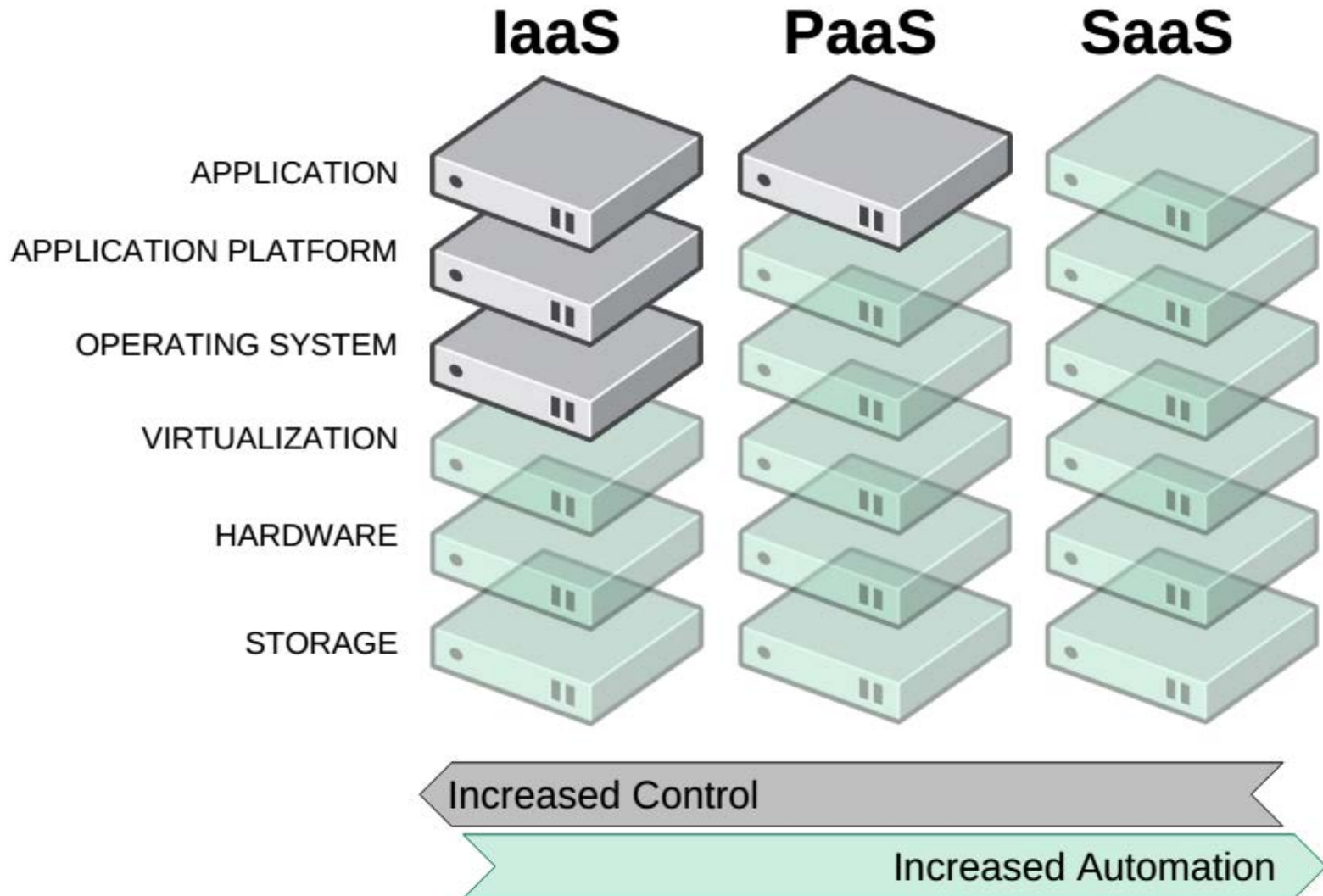


Software Defined Infrastructure

SDI is a way of managing data centers that enables **choice, flexibility, and reduced cost** across the data center's resources (servers, networks, storage), with control of this infrastructure entirely **automated by software**



De Cloud in Laagjes



Wat doet aPaaS?

Wat doet een developer liever niet?

Wat moet er bij elk project opnieuw gebeuren?

Hoe helpt IaaS ons reeds?



'Officiële' Definitie

Application infrastructure (**middleware**),
enriched with cloud characteristics and offered
as a service, is PaaS





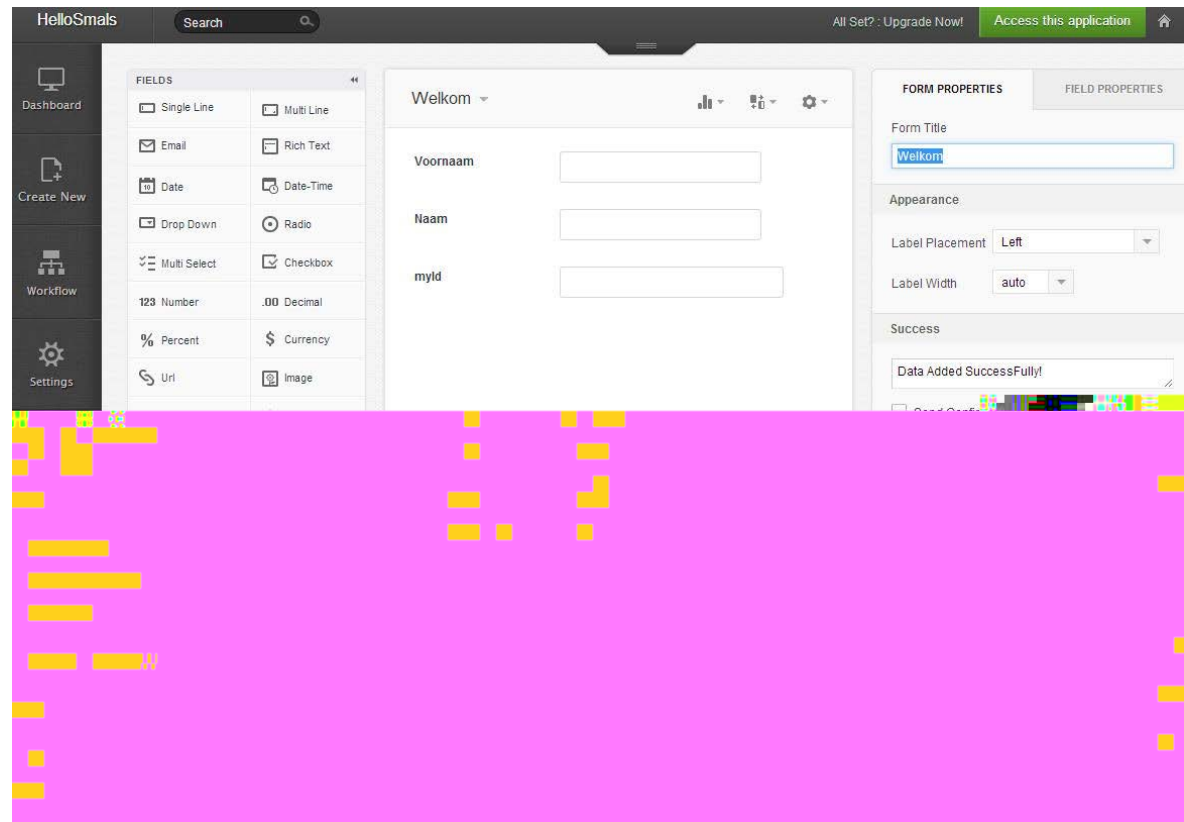
« Zero Coding »

Zoho Creator online

Demo

Aanmaken nieuwe applicatie Zoho Creator

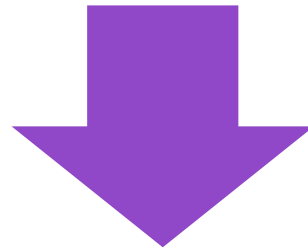
- In te vullen Formulier maken
- Landingspagina maken
- Workflow
- Gebruiker toevoegen





Benefits

Benefits



Gecontroleerde Self-Service

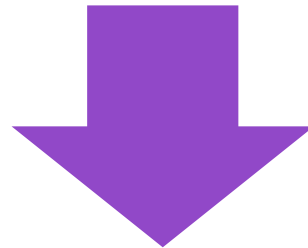
(Developer krijgt standaard resources en kan direct deployen op platform)



Time To Market



Benefits



Makkelijker Migreren

(Geautomatiseerd Migratieproces; identieke omgevingen zijn compatibel)

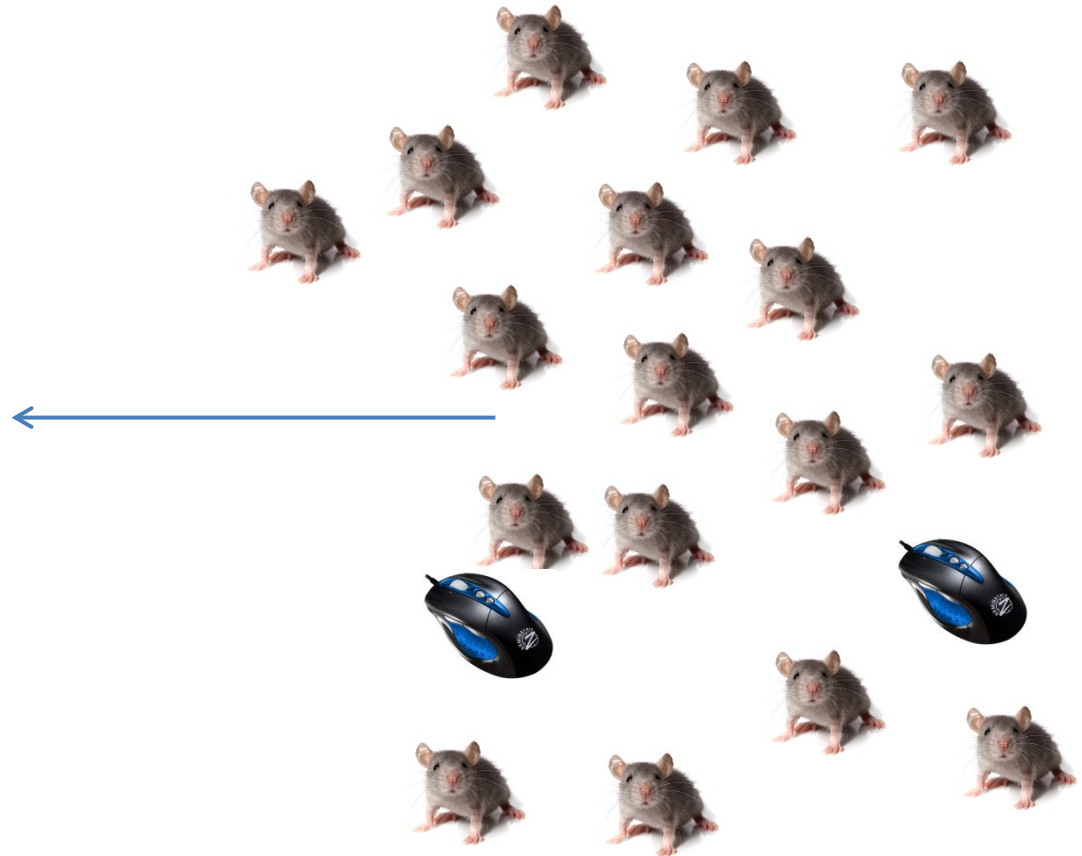


Time To Market

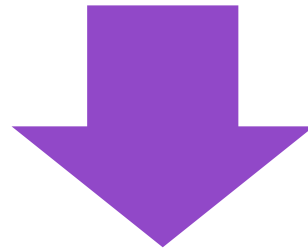
Cost



Elastisch Horizontaal Schalen



Benefits



Elastisch Horizontaal Schalen

(Automatisch meer/minder standaard resources toekennen bij hogere/lagere load)



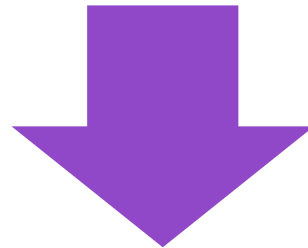
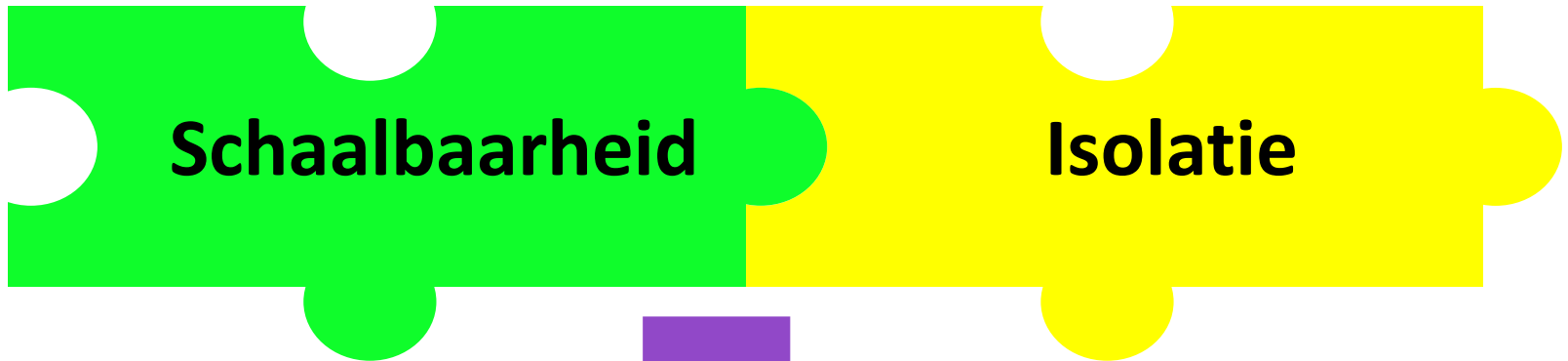
SLA ↑

Cost

(Minder last van pieken in gebruik) (Efficiënter verbruik)



Benefits



Redundantie & Beschikbaarheid

(Gestandaardiseerde resources zijn redundant en beïnvloeden elkaar niet)



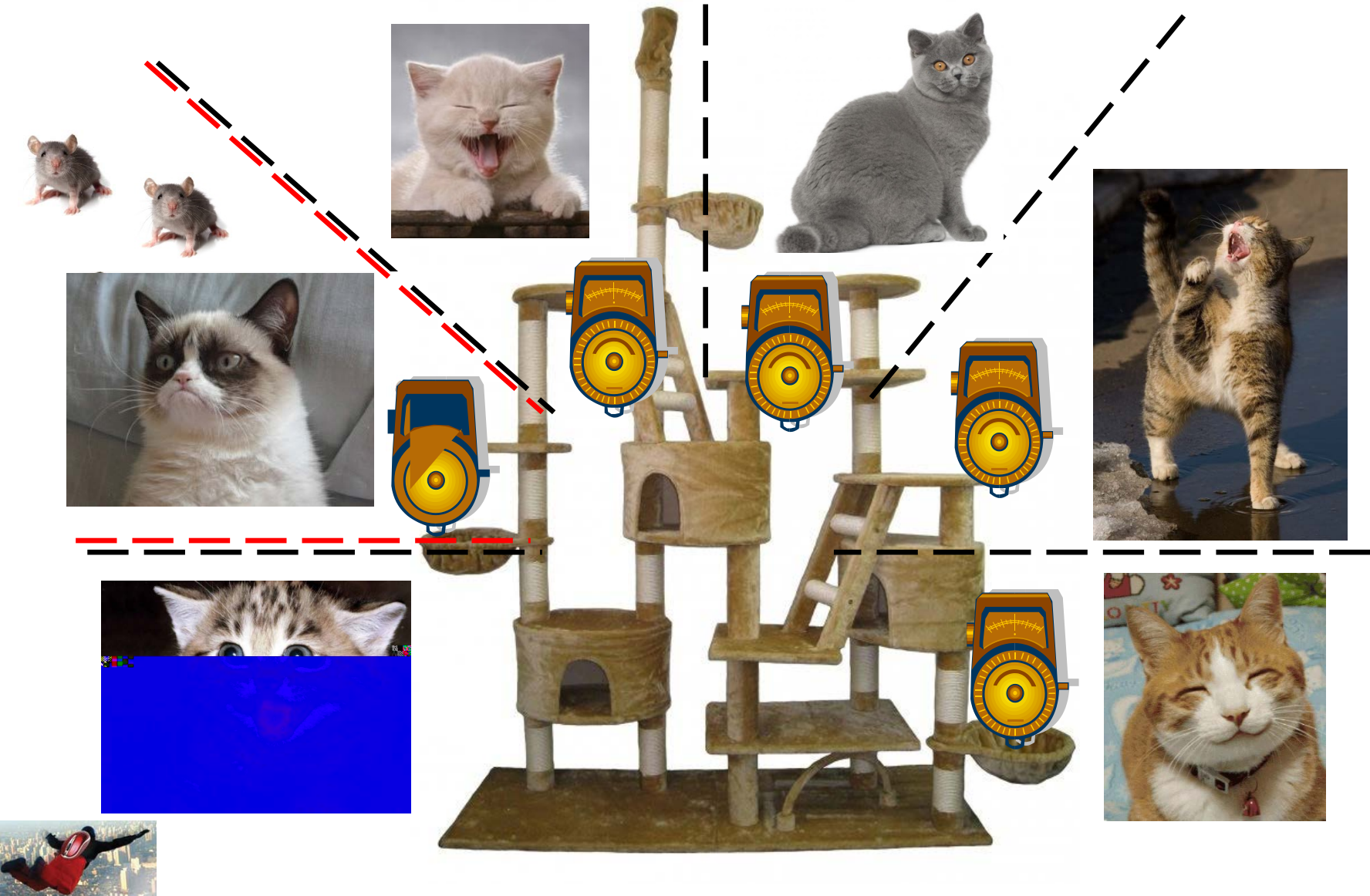
SLA ↑

Cost

(Lager onderhoud)



Multitenancy



Benefits

Schaalbaarheid

Isolatie

Multi-Tenancy

(Van
developers /
applicaties)

Meetbaarheid & Overzichtelijkheid

(Men monitort één groot platform, met 'fine-grained' metingen per applicatie)

SLA ↑

Cost

(Pay per Use Mogelijk)

(Lager onderhoud)



Benefits

- Gecontroleerde Self-Service
- Gemakkelijker Migreren
- Elastisch Schalen
- Redundantie, Beschikbaarheid
- Multitenancy



Time To Market

SLA ↑

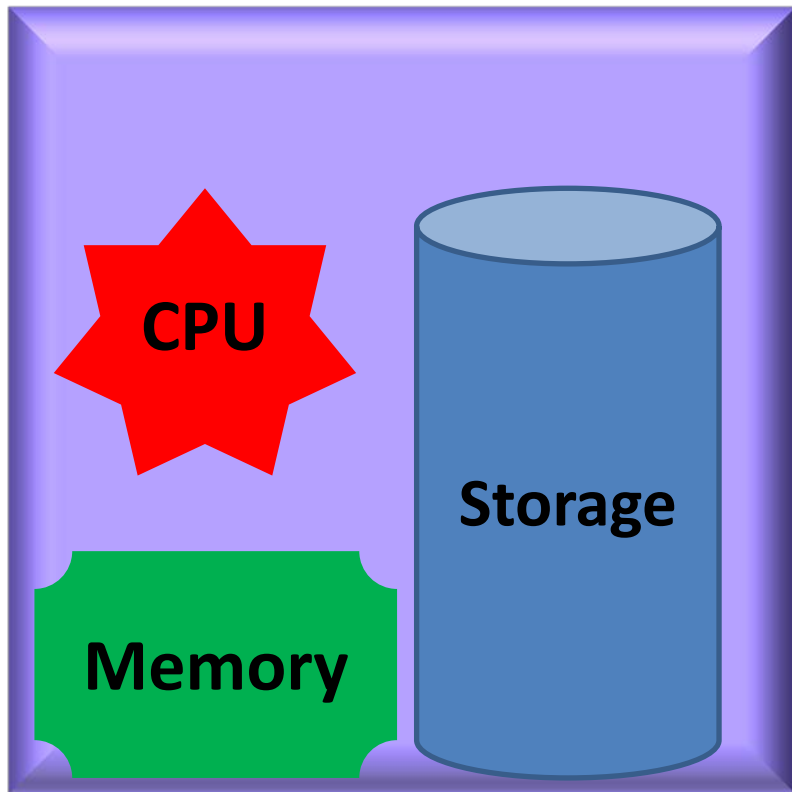
Cost





Interne werking

Wat zijn gestandaardiseerde « Resources » ?



- Abstract, Los van Infrastructuur
- Repliceerbaar
- Identiek (slechts een paar soorten)
- Geïsoleerd
- Gedefinieerd en beheerd door aPaaS

« Software Defined Infrastructure »



v.b. Red Hat OpenShift: « gears »

Public Cloud: Openshift Online



What's Inside a Gear?

Size	Memory	Storage
Small	512 MB	1GB expandable
Medium	1GB	1GB expandable
Medium (with JBoss EAP 6) Java EE6 Full Profile	1GB	1GB expandable
Large <i>Available December 2013</i>	2GB	1GB expandable



v.b. Red Hat OpenShift: « gears »

Public Cloud: Openshift Online



What's Inside a Gear?

Size	Cost
Small	\$0.02/hour after the first 3
Medium	\$0.05/hour
Medium (with JBoss EAP 6) Java EE6 Full Profile	\$0.08/hour
Large <i>Available December 2013</i>	\$0.10/hour

Silver Plan

Professional Red Hat Support

Free Storage upgrades

Access to Extra Resources

\$20 per month

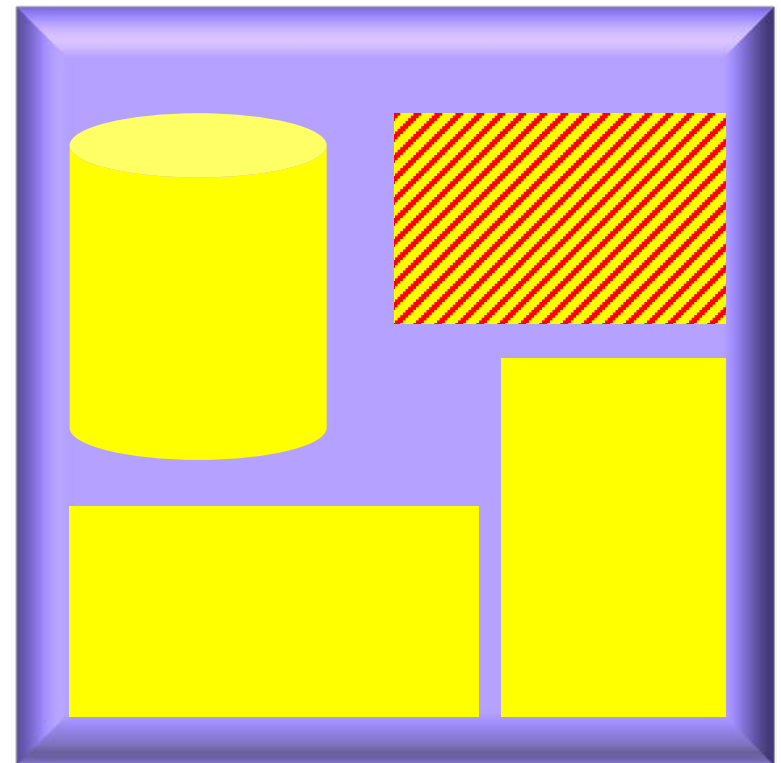
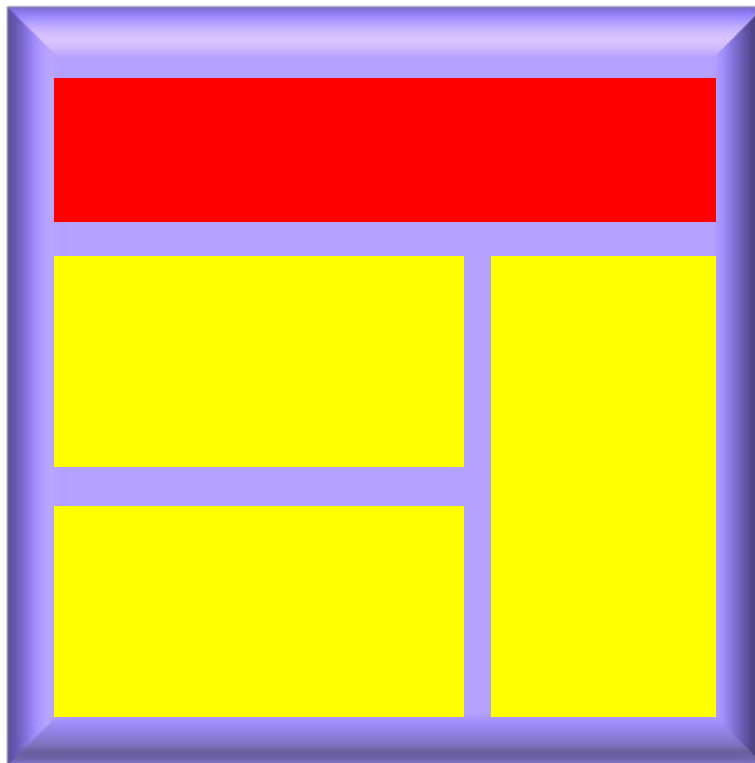
+ usage costs after 3 small ge



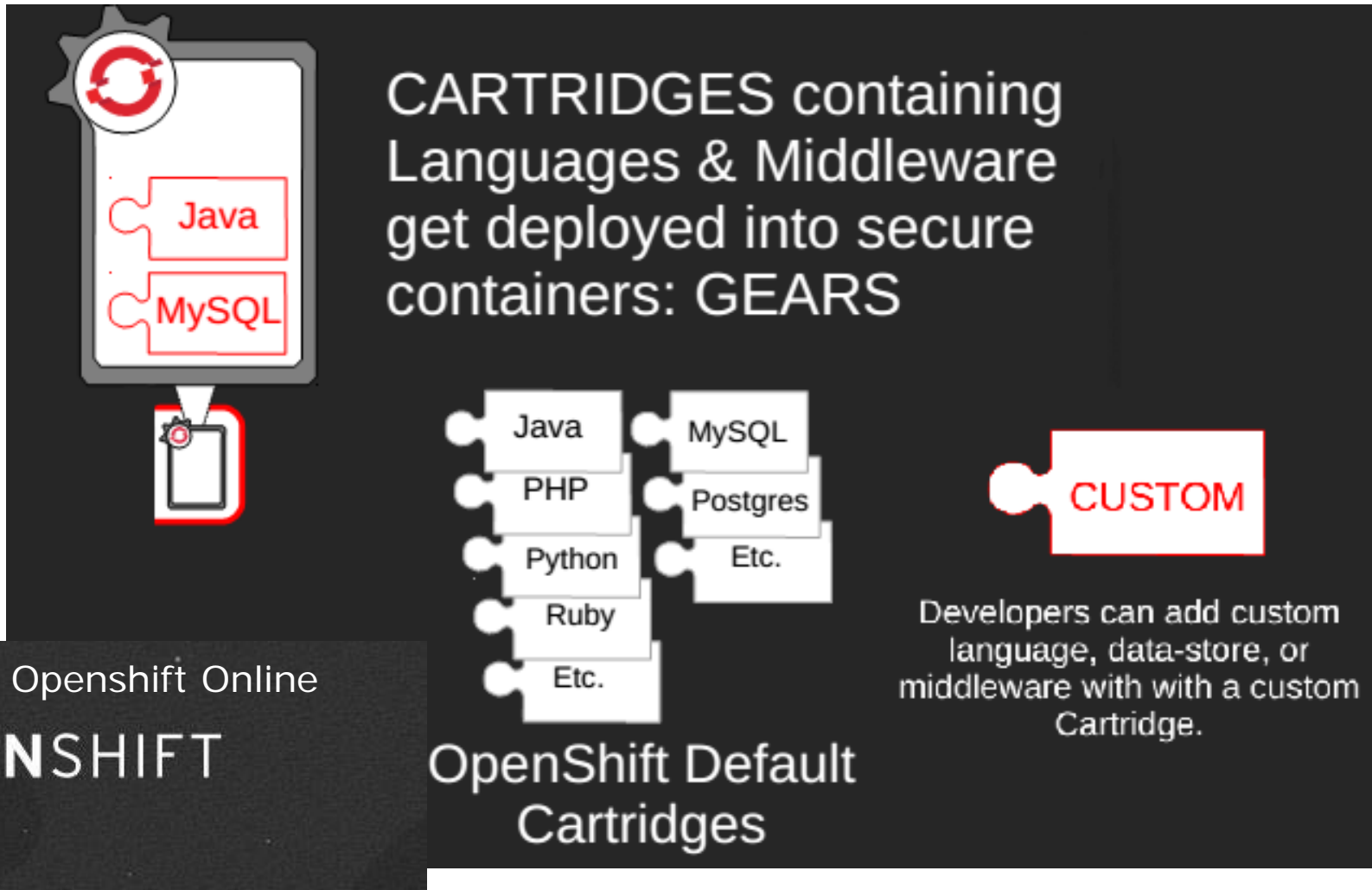
Wat zit er in zo'n « blokje » ?

De diensten die een applicatie nodig heeft, plus de code, verdeeld over repliceerbare delen

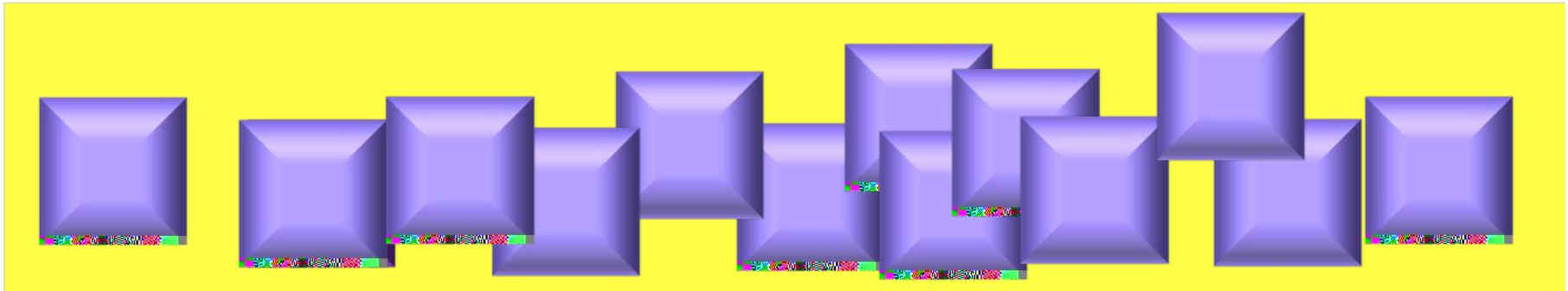
Eg:



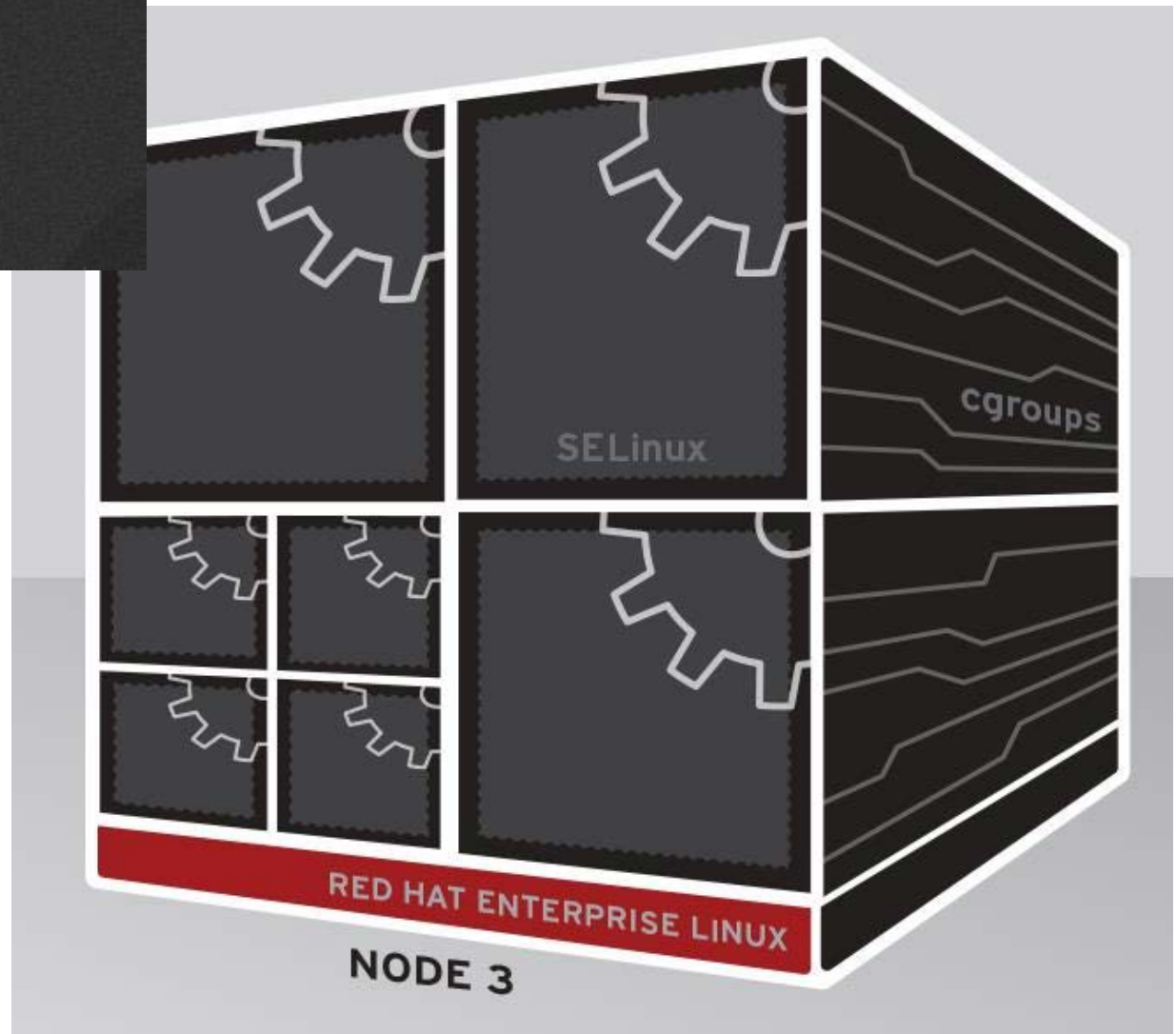
v.b. OpenShift: « cartridges »



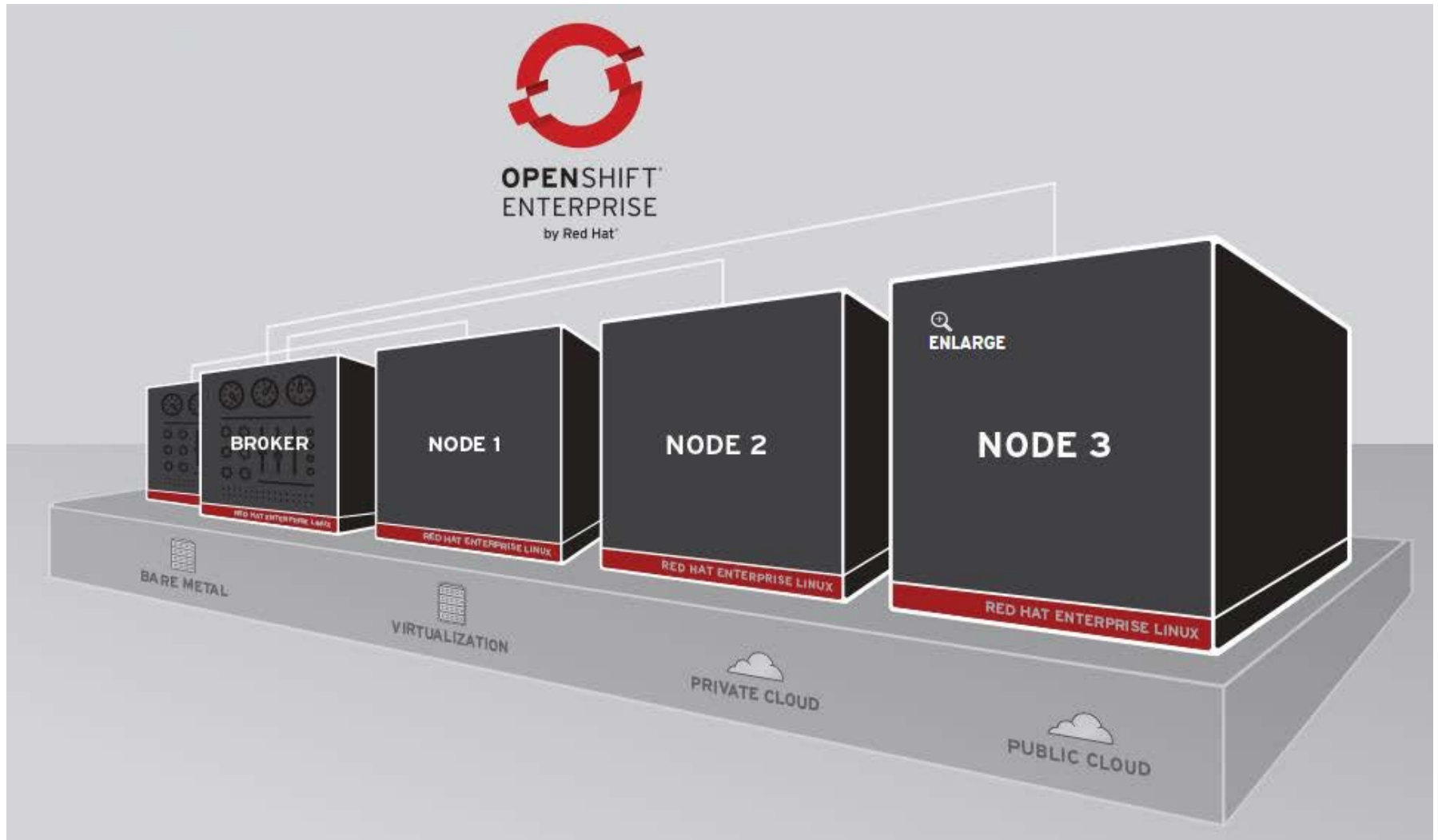
Hoe krijgen de 'blokjes' infrastructuur?



v.b. OpenShift: « nodes »

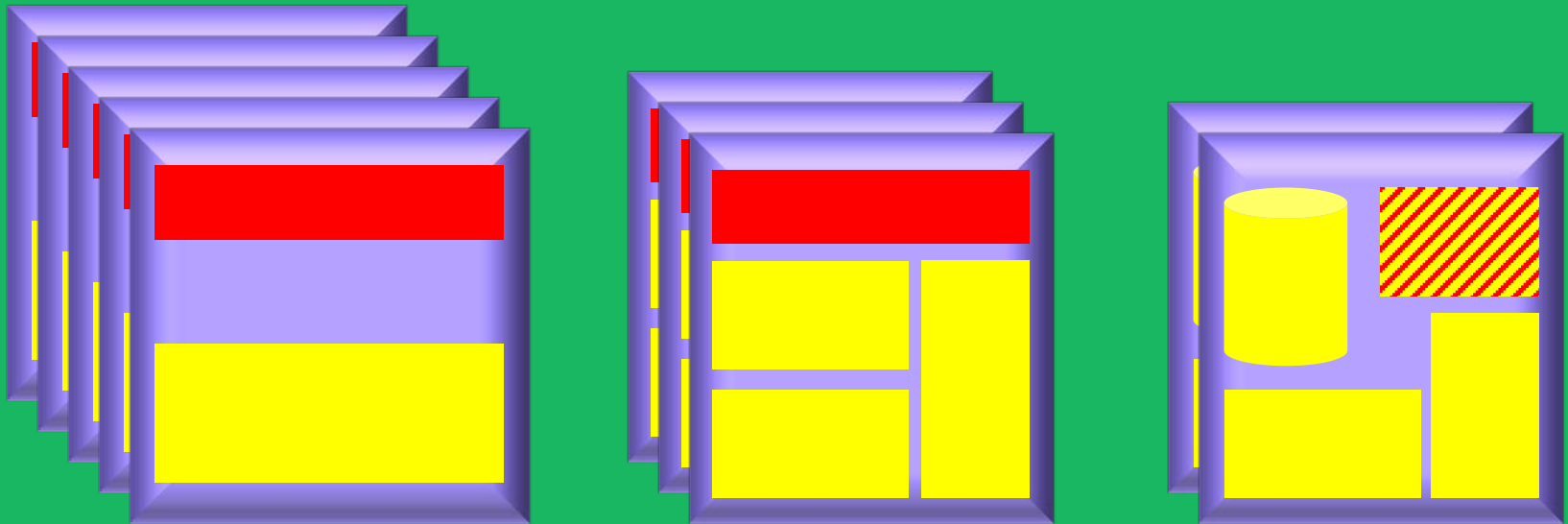


v.b. OpenShift: « broker », enterprise platform

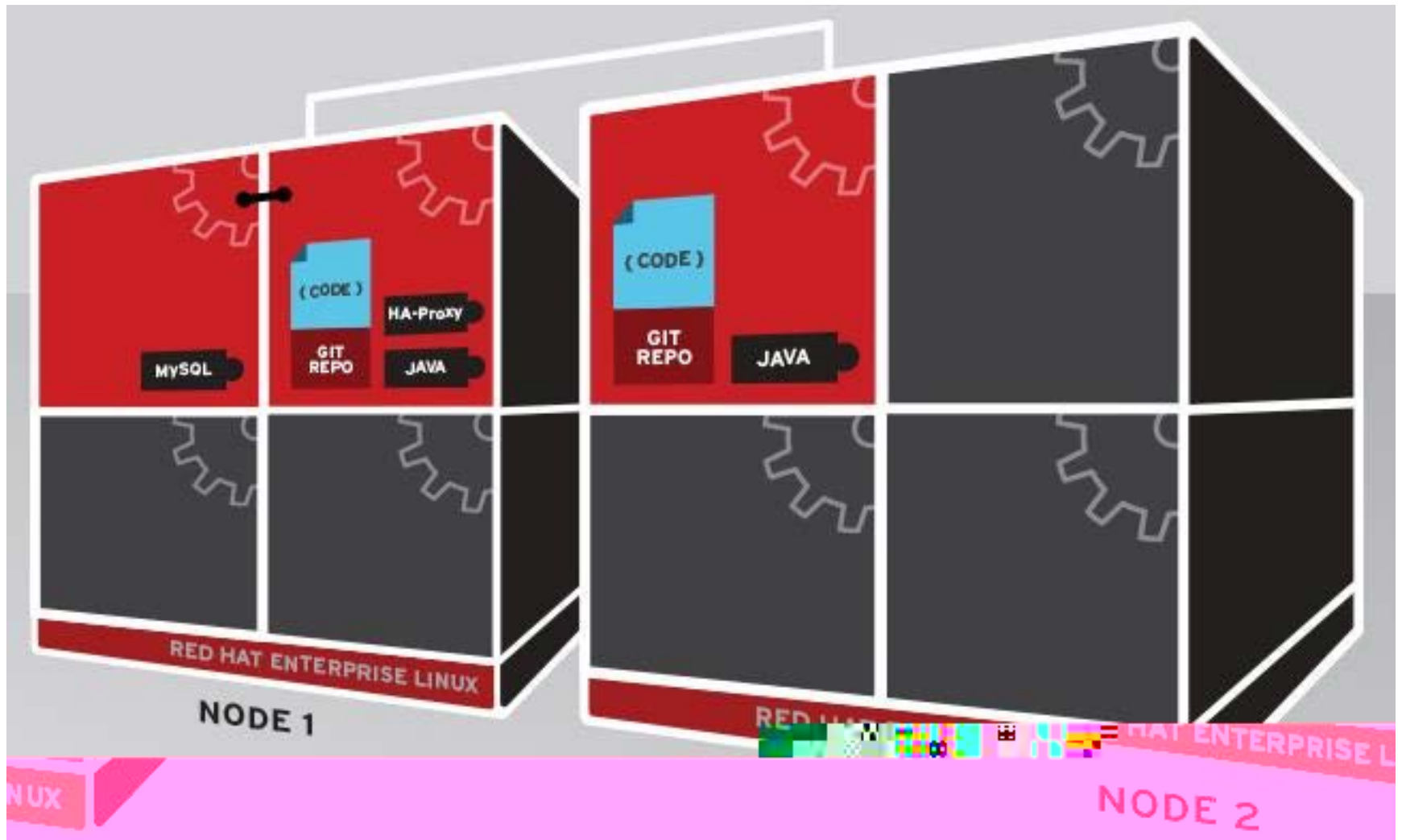


Een VB: 3-tier applicatie, geschaald

Web App



Hoe werkt een aPaaS?



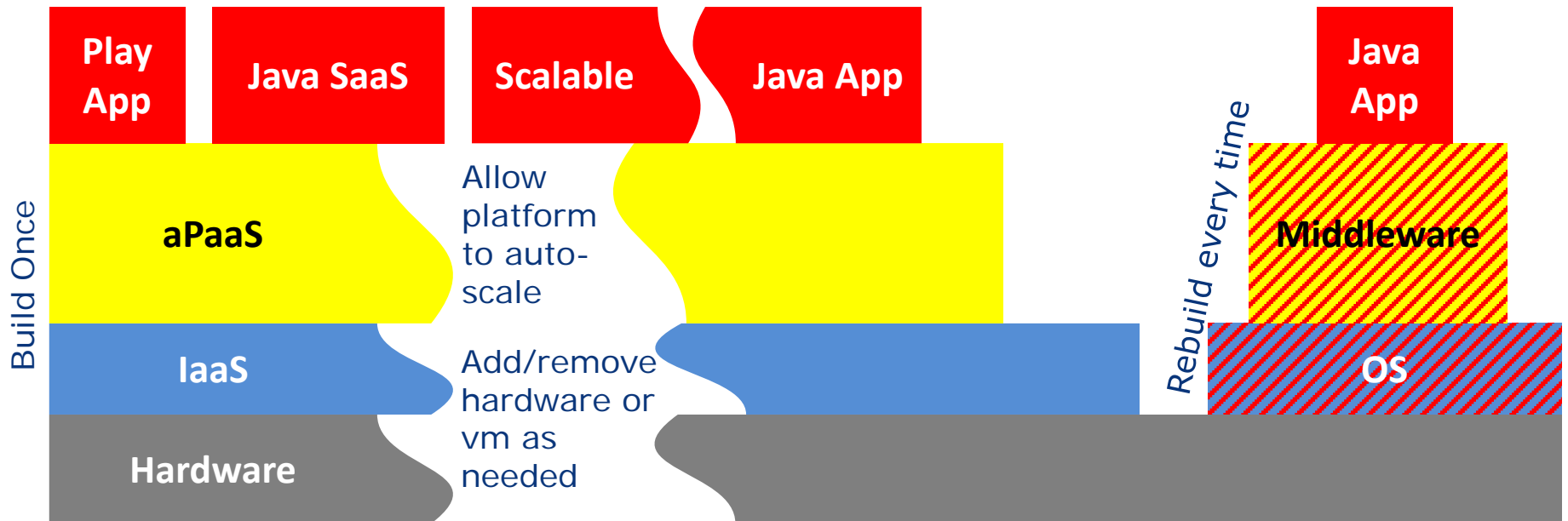
Bouwen op Fundamenten

OS > MiddleWare > App

IaaS > (a)PaaS > SaaS

X-aaS Stack

Traditional Stack



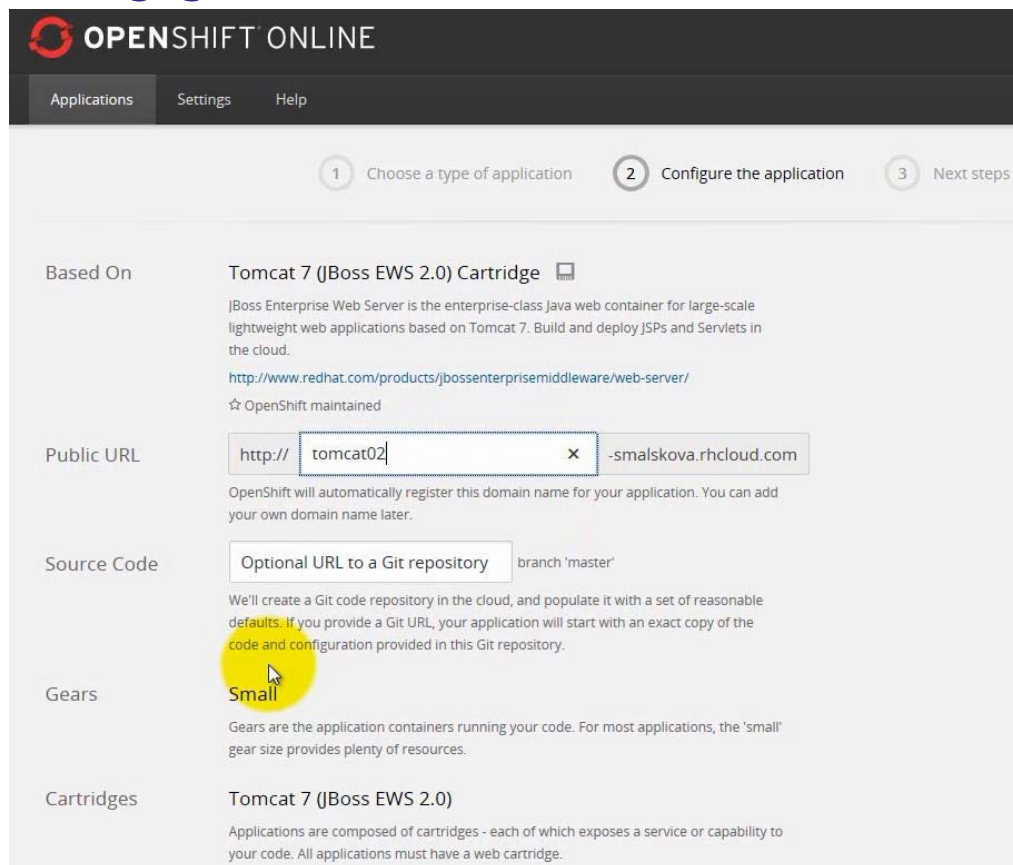


Tomcat servlet
OpenShift Online

Demo

Aanmaken nieuwe applicatie Red Hat Openshift

- « Geef mij een willekeurig getal »
- Tomcat app maken
- Code downloaden
- Code aanpassen
- Code « pushen »



OPENSIFT ONLINE

Applications Settings Help

1 Choose a type of application 2 Configure the application 3 Next steps

Based On Tomcat 7 (JBoss EWS 2.0) Cartridge

JBoss Enterprise Web Server is the enterprise-class Java web container for large-scale lightweight web applications based on Tomcat 7. Build and deploy JSPs and Servlets in the cloud.

<http://www.redhat.com/products/jbossenterprisemiddleware/web-server/>

☆ OpenShift maintained

Public URL

OpenShift will automatically register this domain name for your application. You can add your own domain name later.

Source Code branch 'master'

We'll create a Git code repository in the cloud, and populate it with a set of reasonable defaults. If you provide a Git URL, your application will start with an exact copy of the code and configuration provided in this Git repository.

Gears **Small**

Gears are the application containers running your code. For most applications, the 'small' gear size provides plenty of resources.

Cartridges Tomcat 7 (JBoss EWS 2.0)

Applications are composed of cartridges - each of which exposes a service or capability to your code. All applications must have a web cartridge.





Pauze

Agenda

1. Inleiding
2. Benefits
3. Interne werking



4. **Eigenschappen**

5. Marktoverzicht
6. Impact op taken
7. Conclusies





Eigenschaften

Eigenschappen

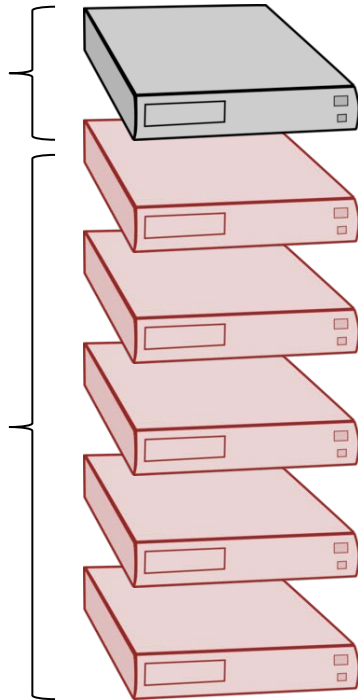
aPaaS-oplossingen zijn er in verschillende smaken en vormen

- **Deployment modellen**
Public vs private vs hybrid
- **Ontwikkeling**
Traditionele ontwikkeling vs zero-coding
- **Talen en frameworks**
Single vs multi-language
- **Cloud features**
Cloud-native vs cloud-based
- **Buy vs build**

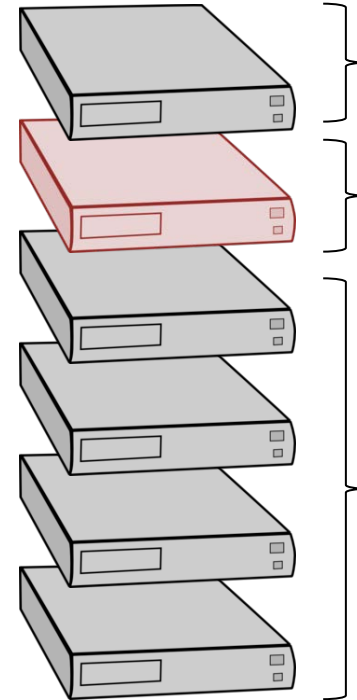


Deployment modellen

Public Cloud



Private Cloud



Deployment modellen

- **Private cloud** deployment belangrijk vanwege vertrouwelijkheid van gegevens
- **Cloud portability**: applicaties kunnen met beperkte effort gemigreerd worden tussen cloud-omgevingen
- **Hybride modellen** mogelijk, mogelijk interessant voor:
 - Cloud bursting: pieken opvangen in de public cloud
 - Experimenteren in public cloud, deployen in private cloud
 - Load testen

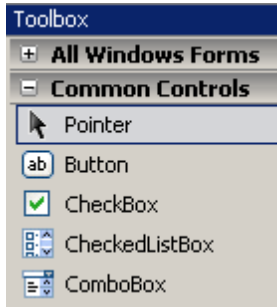


Ontwikkeling

- Oplossingen op basis van traditionele ontwikkeling
 - JEE
 - klassieke IDE
 - Hoge mate van controle
- Zero-coding oplossingen



Zero-coding oplossingen





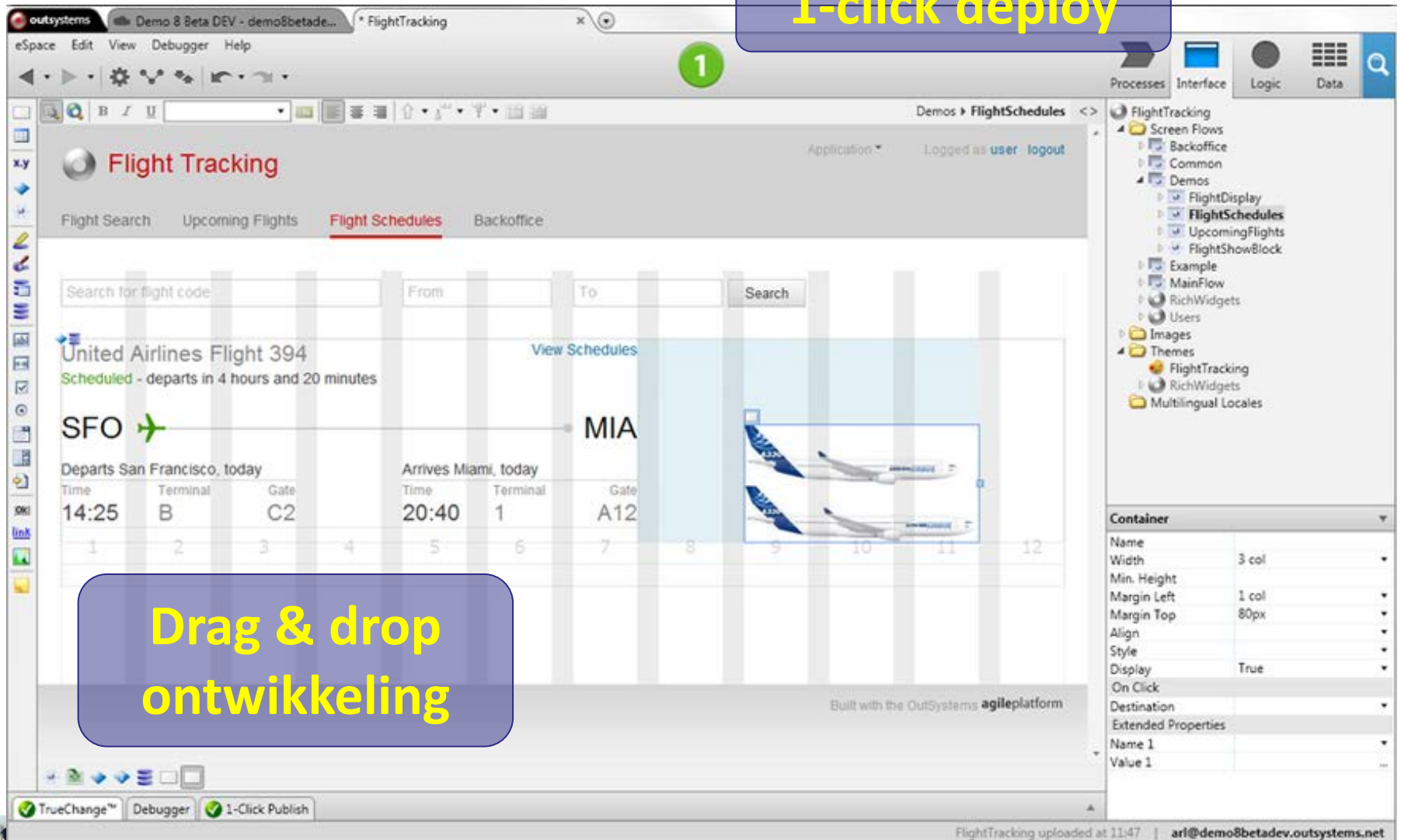
```
$.each(a, function() {
  if (o[this.name] !==
    if (!o[this.name]
      o[this.name]
    }
  }
```





Vb: Outsystems

1-click deploy



Flight Tracking

Flight Search Upcoming Flights **Flight Schedules** Backoffice

Search for flight code From To Search

United Airlines Flight 394
Scheduled - departs in 4 hours and 20 minutes

SFO → MIA

Departs San Francisco, today Arrives Miami, today

Time	Terminal	Gate	Time	Terminal	Gate
14:25	B	C2	20:40	1	A12

Container

Name	
Width	3 col
Min. Height	
Margin Left	1 col
Margin Top	80px
Align	
Style	
Display	True
On Click	
Destination	
Extended Properties	
Name 1	
Value 1	

TrueChange™ Debugger 1-Click Publish

FlightTracking uploaded at 11:47 | arl@demo8betadev.outsystems.net

Talen en frameworks

- Oorspronkelijk eerder single language oplossingen
- Tendens naar multi-language (= polyglot)
- Voorbeeld: Apprenda: naast .NET nu ook Java
- Zelfde workflow automatisatie en tools voor verschillende talen en frameworks
- Idem voor databases



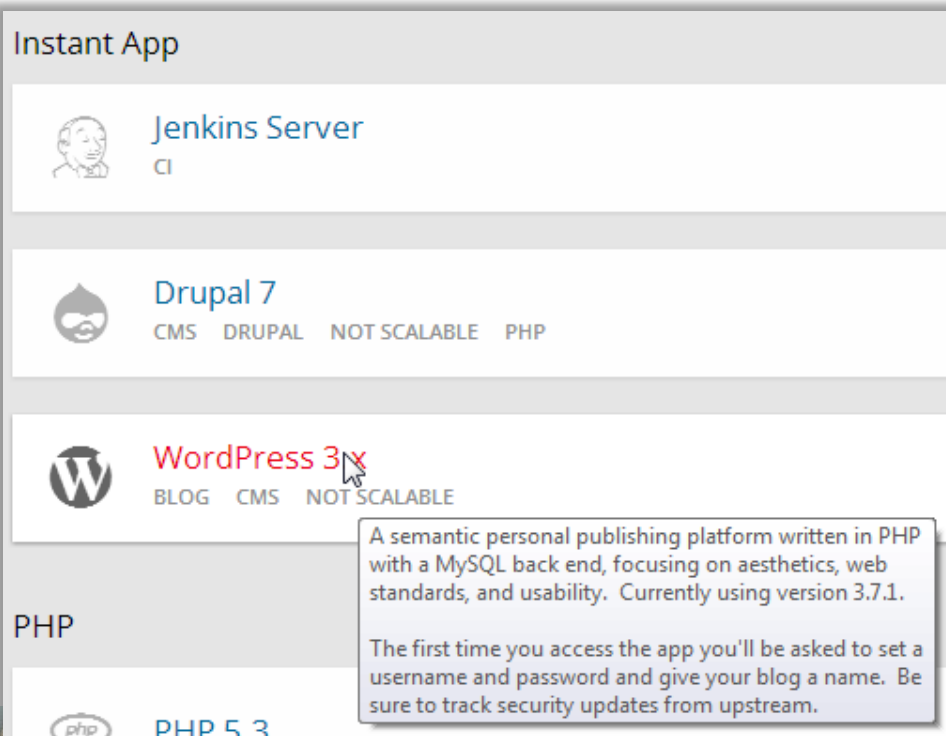


OpenShift


« Instant Apps »


Demo Talen en frameworks


- Mogelijk om 3rd party (open source) software pakketten te deployen
- Vb: Openshift "Instant Apps" (Drupal, WordPress)



Instant App

 **Jenkins Server**
CI

 **Drupal 7**
CMS DRUPAL NOT SCALABLE PHP

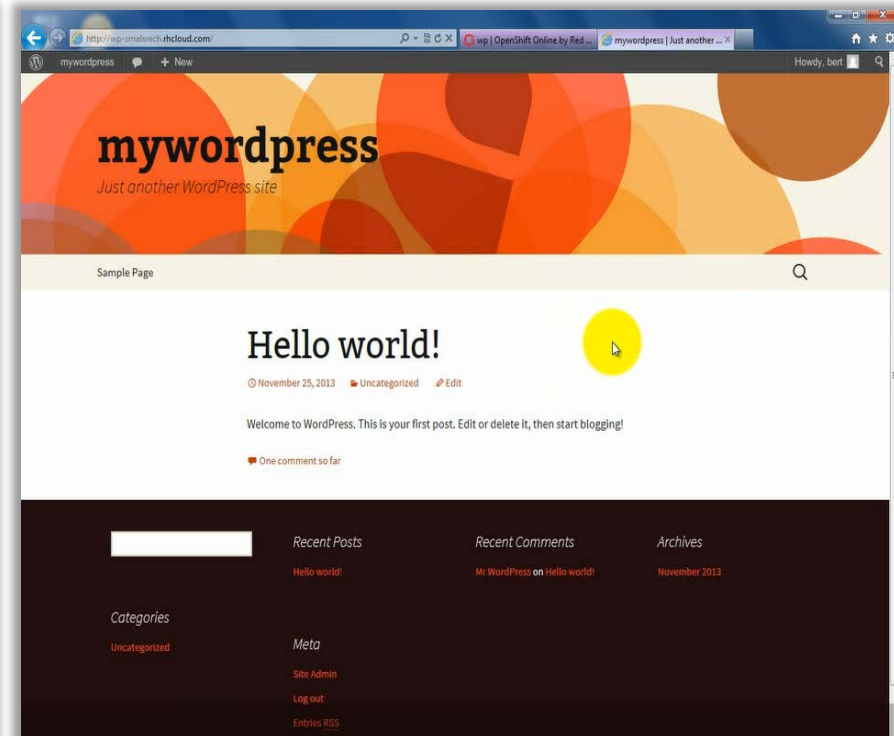
 **WordPress 3.x**
BLOG CMS NOT SCALABLE

PHP

PHP 5.3

A semantic personal publishing platform written in PHP with a MySQL back end, focusing on aesthetics, web standards, and usability. Currently using version 3.7.1.

The first time you access the app you'll be asked to set a username and password and give your blog a name. Be sure to track security updates from upstream.



mywordpress
Just another WordPress site

Sample Page

Hello world!

November 25, 2013 Uncategorized Edit

Welcome to WordPress. This is your first post. Edit or delete it, then start blogging!

One comment so far

Recent Posts
Hello world!

Recent Comments
Mr WordPress on Hello world!

Archives
November 2013

Categories
Uncategorized

Meta
Site Admin
Log out
Entries RSS

Cloud features

- **Cloud-based**
 - Bestaande middleware stacks aangeboden in de cloud, bovenop IaaS
 - VM's niet zichtbaar voor aPaaS-gebruiker
 - Klassieke middleware software zonder specifieke cloud-eigenschappen
- **Cloud native**
 - VM's niet zichtbaar voor aPaaS-gebruiker
 - Cloud-aware middleware



Buy vs Build

- Buy



- Geïntegreerde "aPaaS in a box" oplossing, dikwijls gebaseerd op open source framework
- Vb: Activestate Stackato, gebaseerd op Cloud Foundry framework
- Betalende support, SLA's

- Build



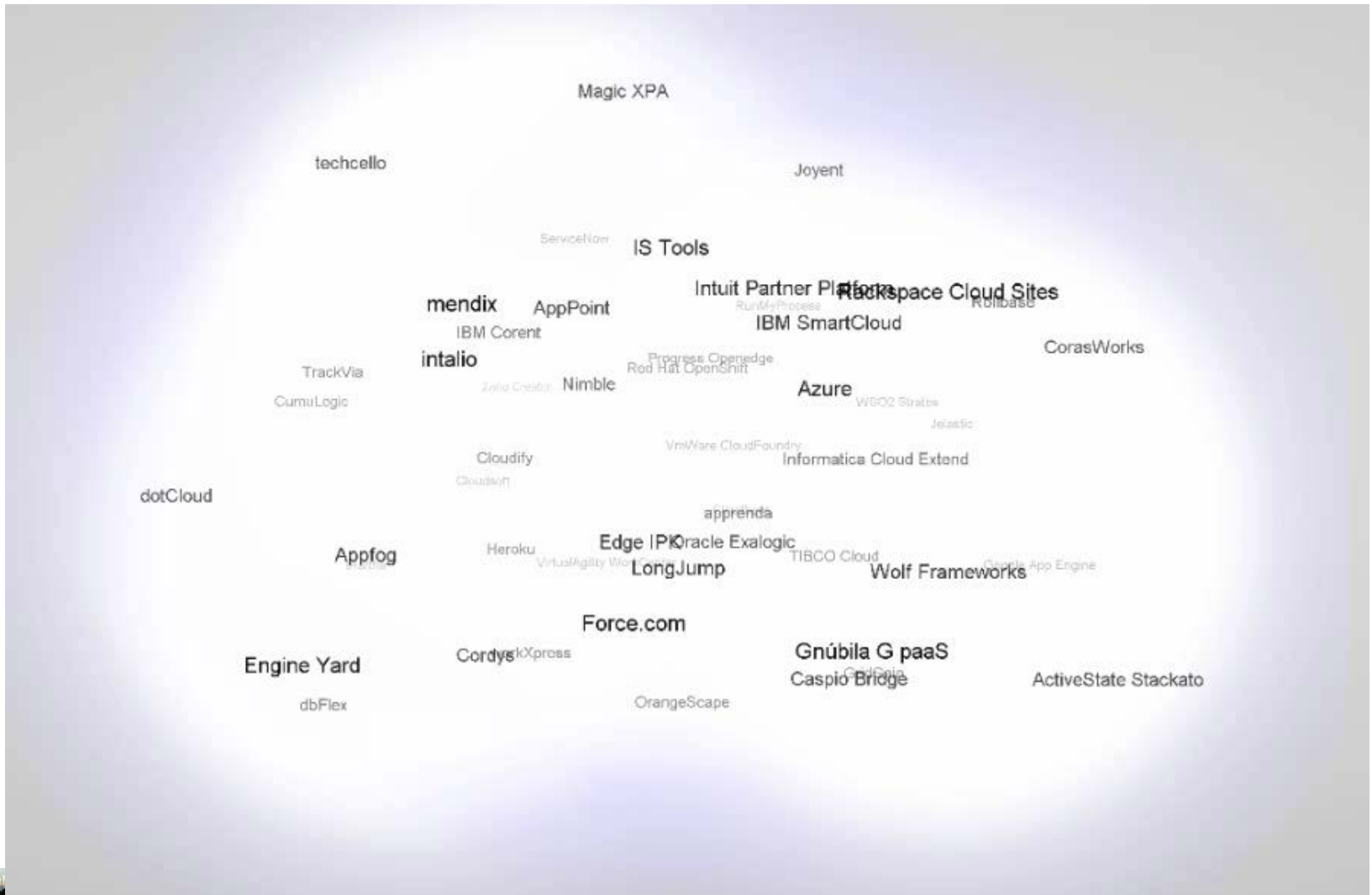
- Zelf een oplossing bouwen op basis van een (open source) framework
- Al snel customisatie nodig → meer lock-in
- Community support





Marktoverzicht

Marktoverzicht



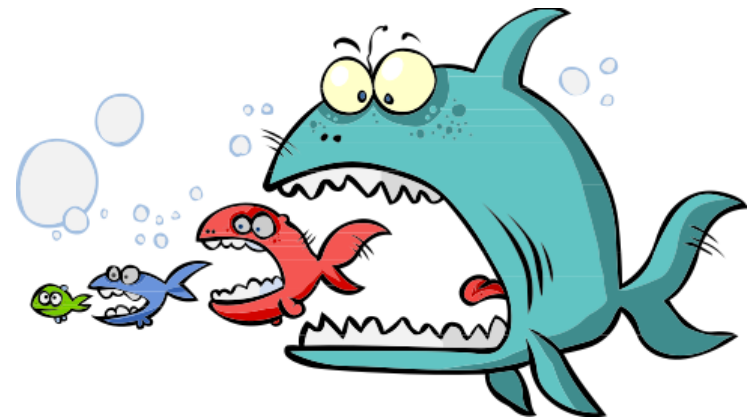
Verschillende invalshoeken

- Leveranciers van **applicatie-infrastructuur** (bvb. Oracle, IBM, ...)
- Leveranciers met focus op **infrastructuur-automatisatie** (bvb. Gigaspaces Cloudify)
- **SaaS-leveranciers** die onderliggend platform als aPaaS aanbieden (bvb. Salesforce Force.com)
- Pure public cloud, **zero-coding** oplossingen (bvb. Cordys)



Consolidatie

- Heel wat **overnames** in de afgelopen maanden
- Wijst op **immature markt**



Wat leveren de grote vendors?



- Maakt deel uit van de Cloud Foundry community
- WebSphere Liberty Buildpack voor Cloud Foundry (WebSphere Application Server Liberty Core)



- Windows Azure platform
- Private cloud: gebaseerd op Windows Server en System Center



Wat leveren de grote vendors?

ORACLE®

- Oracle Cloud Platform
 - Oracle Database (as a service)
 - Oracle Cloud Application Foundation (Weblogic, ...)
 - Oracle Enterprise Manager
 - Kan draaien op Oracle engineered systems (Exalogic, Exadata)
- In de public cloud: Oracle Java Cloud Service, Weblogic as a Service



Open source gebaseerde oplossingen



Apache Stratos

- WSO2 Stratos
- WSO2 StratosLive



Cloud Foundry

- Pivotal
- ActiveState Stackato
- IBM (Websphere Liberty Buildpack)



Red Hat OpenShift Origin

- OpenShift Online
- OpenShift Enterprise









Vendor lock-in

- Open source \neq "geen vendor lock-in"
- Cloud portability \neq "geen vendor lock-in"
 - Cloud portability betekent dat aPaaS-platform onafhankelijk is van onderliggende (cloud-)infrastructuur
- Momenteel geen **application portability**
 - Applicaties kunnen niet zomaar overgedragen worden naar een ander aPaaS-platform
 - Verschillende packaging formaten (Buildpacks, Cartridges)
- Gelijkaardige concepten maakt **overstappen** van de ene oplossing naar de andere eenvoudiger



Zero-coding oplossingen

Oplossing	Public/private cloud?	Highlights
		
		
		
		
		
		



Shortlist

- Meest interessante producten in onze context:



- Vanwege
 - On-premise deployment
 - Traditionele ontwikkeling
 - Open-source gebaseerd
 - Polyglot



Samenvattend overzicht

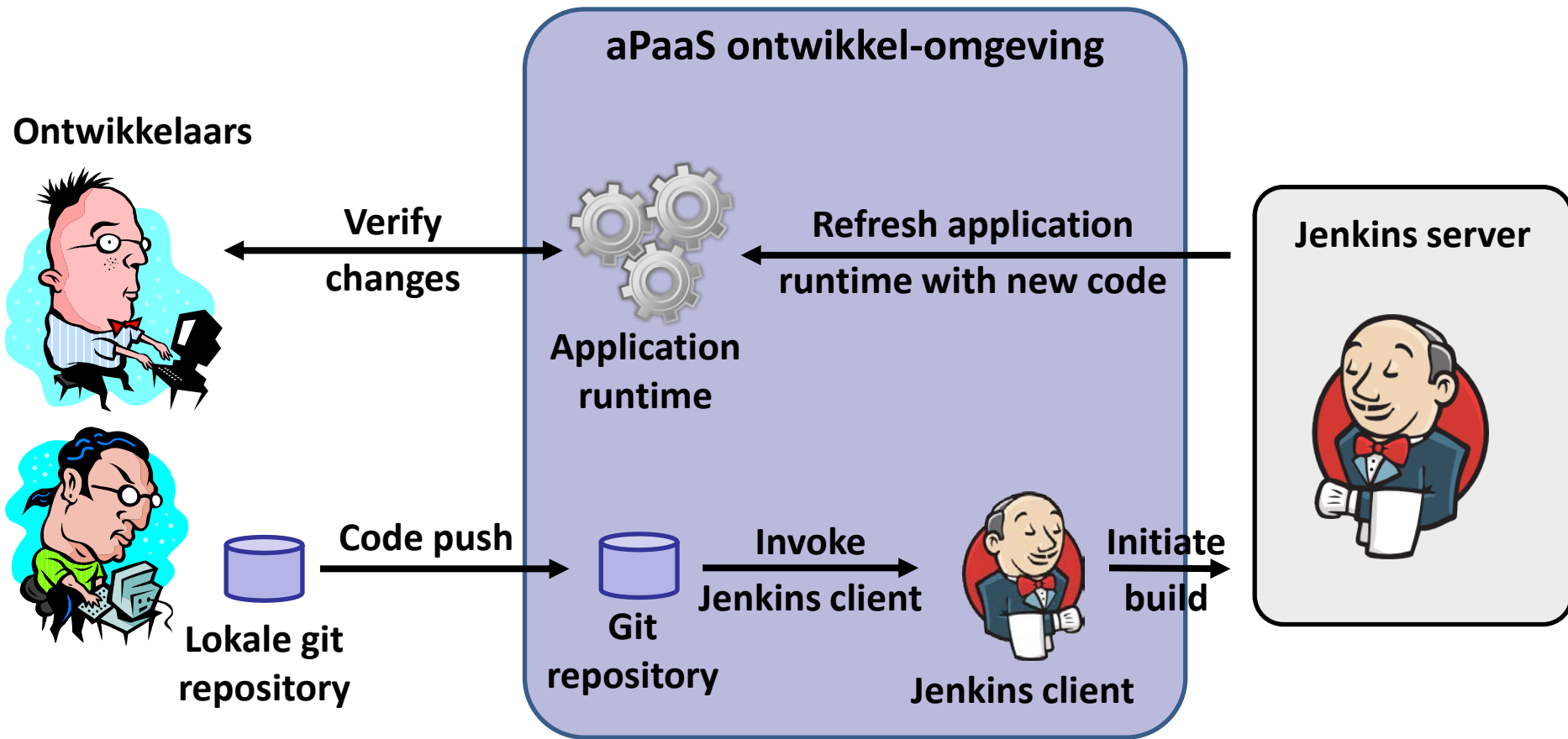
Oplossing	Public/private	Ontwikkeling	Single/multi-language	Cloud-based/native	Buy vs build
			→		





Impact op taken

Developer workflow



Ontwikkeltools

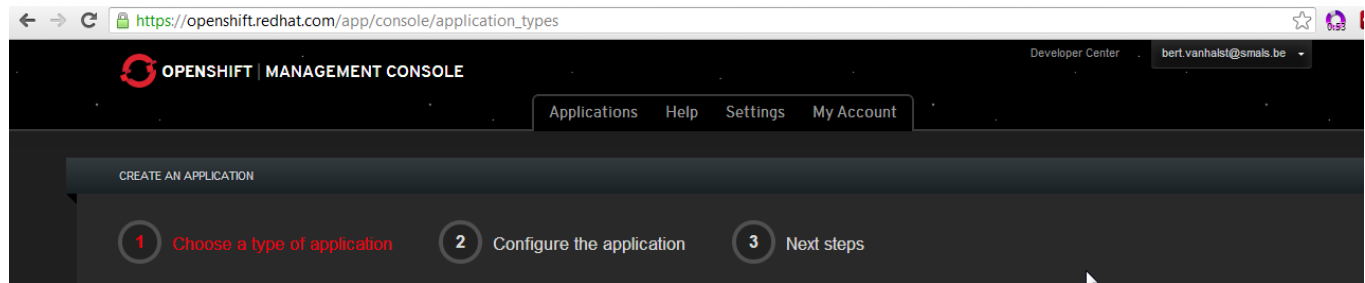


Command-line

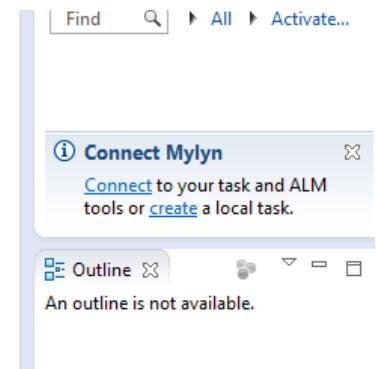
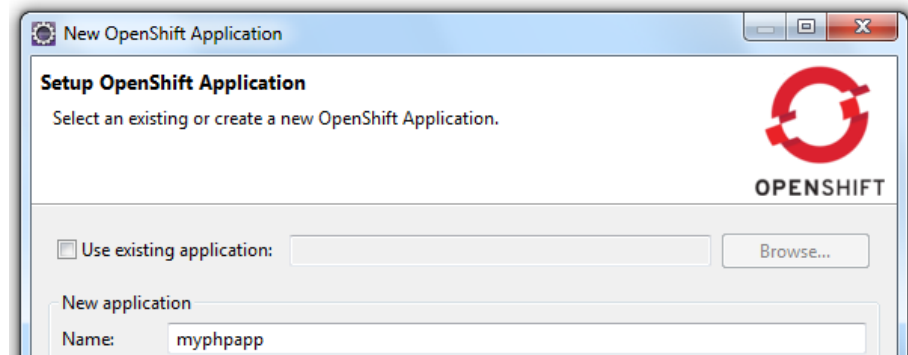
```
C:\Windows\system32\cmd.exe  
Microsoft Windows [Version 6.1.7601]  
Copyright (c) 2009 Microsoft Corporation. All rights reserved.  
C:\Users\beva>rhc app create MyApp jbossas-7
```



Web console



IDE



Ontwikkeltools

- "Don't change a winning team..."
- **Huidige tools** zoveel mogelijk blijven gebruiken
 - IDE: Eclipse
 - SCM: Git / SVN
 - CI: Jenkins
 - Maven
 - Test tools
- Huidige **competenties** inzetbaar
- **Leercurve** beperken



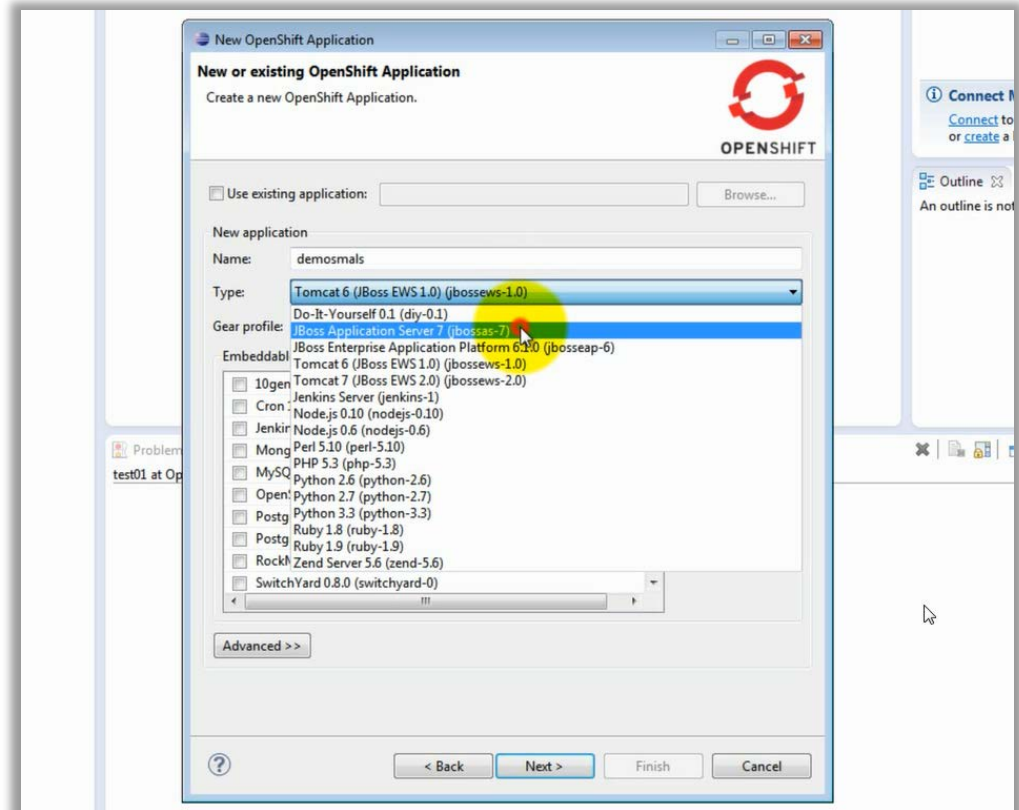


OpenShift Eclipse Plugin

Demo

Aanmaken nieuwe applicatie + code aanpassen

- Eclipse + JBoss OpenShift Tools plugin
- OpenShift Online
- Jenkins



Impact voor ontwikkelaar/architect

- Gecontroleerde **self-service**
 - Automatische provisioning van infrastructuur
 - Uitgebreidere keuze aan talen/frameworks, binnen grenzen van wat ondersteund wordt binnen de organisatie
- **Schalen** van toepassingen
 - Wie beslist over (automatisch) schalen?
 - Applicatie moet erop voorzien zijn



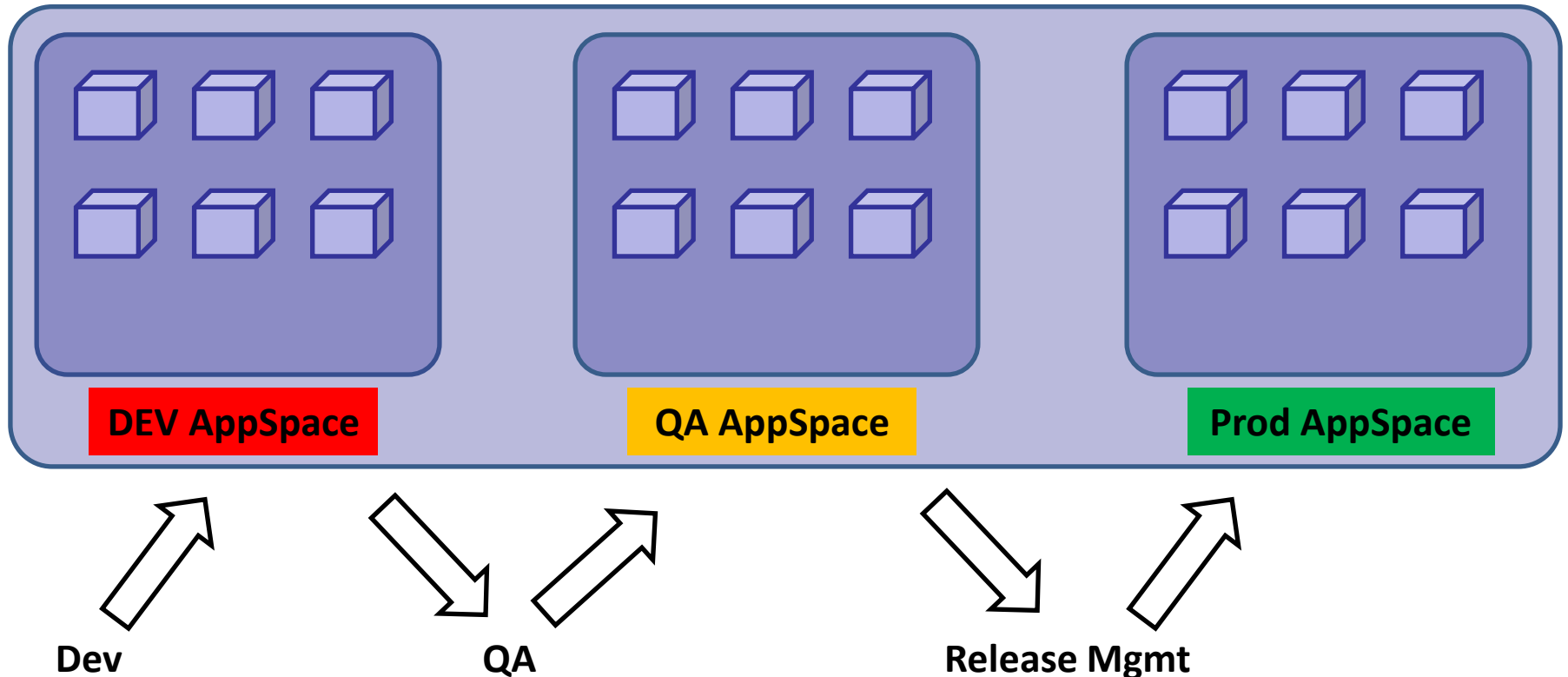
Release management

- Technisch mogelijk om als ontwikkelaar te deployen tot in productie
- Is niet noodzakelijk wat we willen
- Traditionele checkpoints om applicaties te valideren alvorens ze in productie gaan
- Logische of fysieke scheiding van omgevingen
- Toegangsrechten in te stellen volgens rol



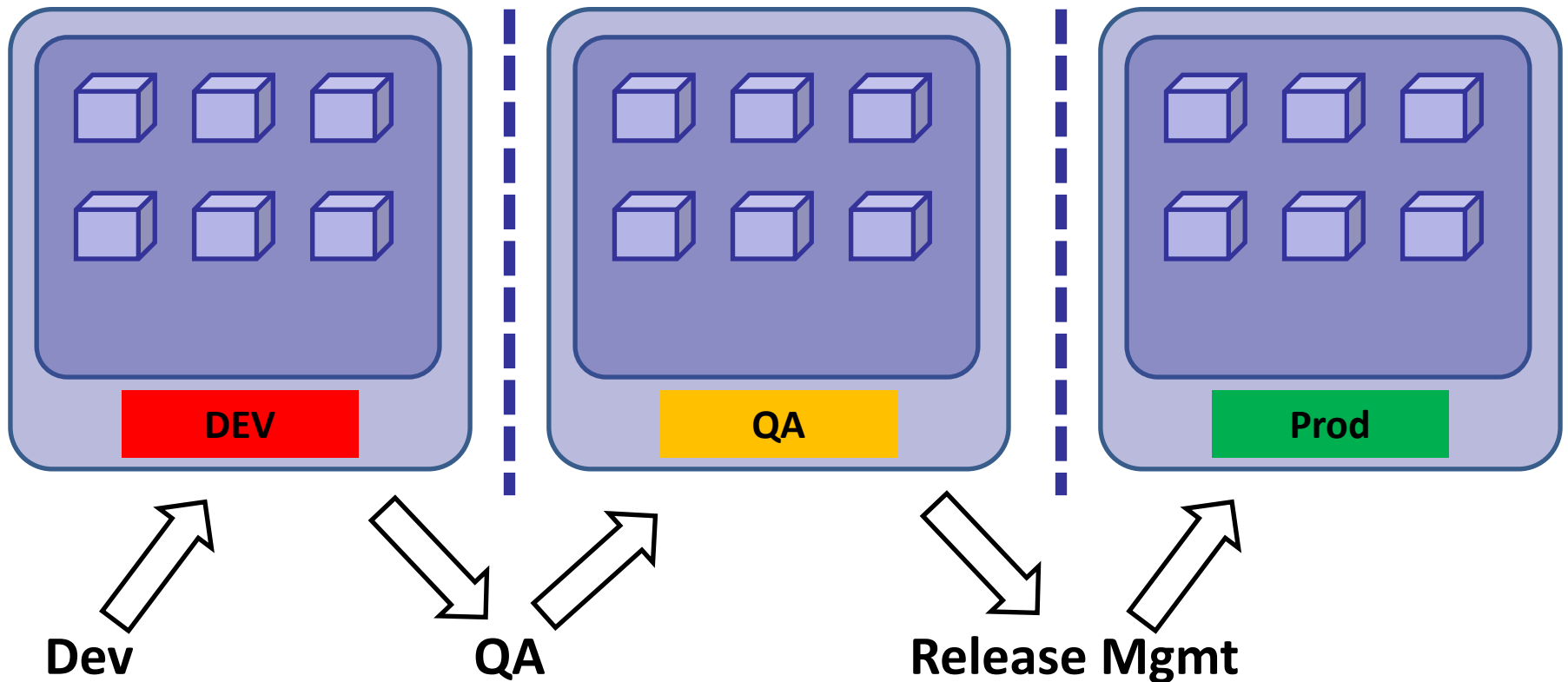
Release management

- **Logische** scheiding van omgevingen
- Eén aPaaS met meerdere omgevingen



Release management

- **Fysische** scheiding van omgevingen
- Apart aPaaS-platform per omgeving



Integratie met bestaande componenten

- Integratie nodig met bestaande componenten bij opzetten aPaaS-omgeving
- Gebruik van bestaande **databases en webservices**
- **Communicatie** tussen toepassingen
- **Gebruikersbeheer** voor authenticatie/autorisatie van gebruikers
- **Monitoring en logging**



Impact op infrastructuurbeheer

- **Initiële inspanning** nodig om aPaaS-platform op te zetten
 - Netwerk-architectuur
 - Bepalen en beheren van onderliggende infrastructuur (bare-metal / VM's / IaaS)
- Automatische provisioning van resources (application stacks) → meer focus op **capacity management**: voorzien extra hardware (op basis van IaaS-platform, virtuele servers)
- **Updates** beheren: infrastructuur, application stacks en aPaaS-platform zelf
 - Impact op SLA's (beschikbaarheid)?

Bottom-line: automatisatie van herhaalbare taken



The image features a background of a wooden pallet, showing the texture and grain of the wood. A white rounded rectangle is positioned in the lower-left area, containing the word "Conclusies" in a bold, purple, sans-serif font.

Conclusies

Samenvatting

- **aPaaS** = application infrastructure as a service
- Beoogde **voordelen**:
 - Kortere time-to-market
 - Lagere kost
 - Verhoogde SLA
- Middels doorgedreven **automatisatie** en **standaardisatie**



Uitdagingen

- Markt nog **niet matuur**
 - Oplossingen zijn nog jong
 - Geen standaarden



Uitdagingen

- Keuze vandaag is **niet noodzakelijk lange termijn** oplossing
- Vandaag **weinig concrete cases**
 - Grotendeels voor niet-kritische toepassingen
 - PayPal, Cisco
- **Integratie** met bestaande systemen



Aanbevelingen

- **Tendens** naar software-defined infrastructure, tot en met applicatie-infrastructuurlaag
- Aangeraden om **aPaaS-kennis op te doen**, te experimenteren
 - Eerste stappen kunnen (gratis) in de public cloud
- **On-premise** deployment aangeraden
- Maar opgelet voor **customisatie**
- **IaaS**: niet nodig, maar aangeraden



aPaaS @Smals?

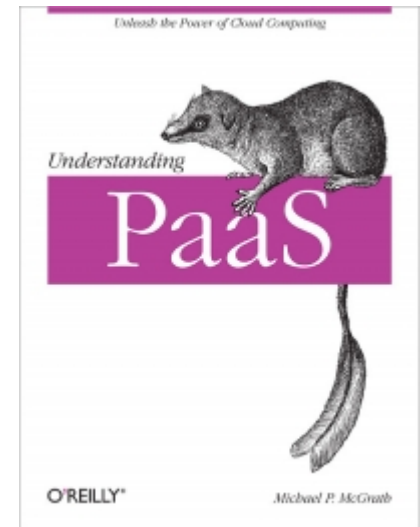
- Tot nu toe:
 - Studie van de **concepten**
 - Uitvoering van **marktonderzoek**
 - Inschatting **impact** op taken voor ontwikkeling en infrastructuurbeheer
 - Voorstel scope **POC**
- In samenwerking met Enterprise Architecten





Blogs / boeken / artikels / ...

- Understanding PaaS, *Michael P. McGrath*
<http://shop.oreilly.com/product/0636920023128.do>
- Cloud Foundry Blog
<http://blog.cloudfoundry.com/>
- Openshift Blog
<https://www.openshift.com/blogs>
- Programming for PaaS, *Lucas Carlson*
<http://shop.oreilly.com/product/0636920026235.do>



Bijlage - producten

- Activestate Stackato - <http://www.activestate.com/stackato>
- Apprenda - <http://apprenda.com/>
- Cloud Foundry - <http://www.cloudfoundry.com/>
- Cordys - <http://www.cordys.com/platform-as-a-service>
- IBM - <http://www.ibm.com/cloud-computing/us/en/paas.html>
- Longjump - <http://www.longjump.com/>
- Mendix - <http://www.mendix.com/>
- Microsoft - http://www.microsoft.com/industry/government/guides/cloud_computing/5-PaaS.aspx
- Oracle - <http://www.oracle.com/us/solutions/cloud/platform/overview/index.html>
- Outsystems - <http://www.outsystems.com/>
- Red Hat Openshift – <https://www.openshift.com/>
- Rollbase - <http://www.progress.com/products/rollbase>
- Salesforce - <http://www.salesforce.com/platform/overview/>
- WSO2 Stratos - <http://wso2.com/cloud/stratos/>
- Zoho - <https://www.zoho.com/creator/>

