Java Development for the Cloud, present and future
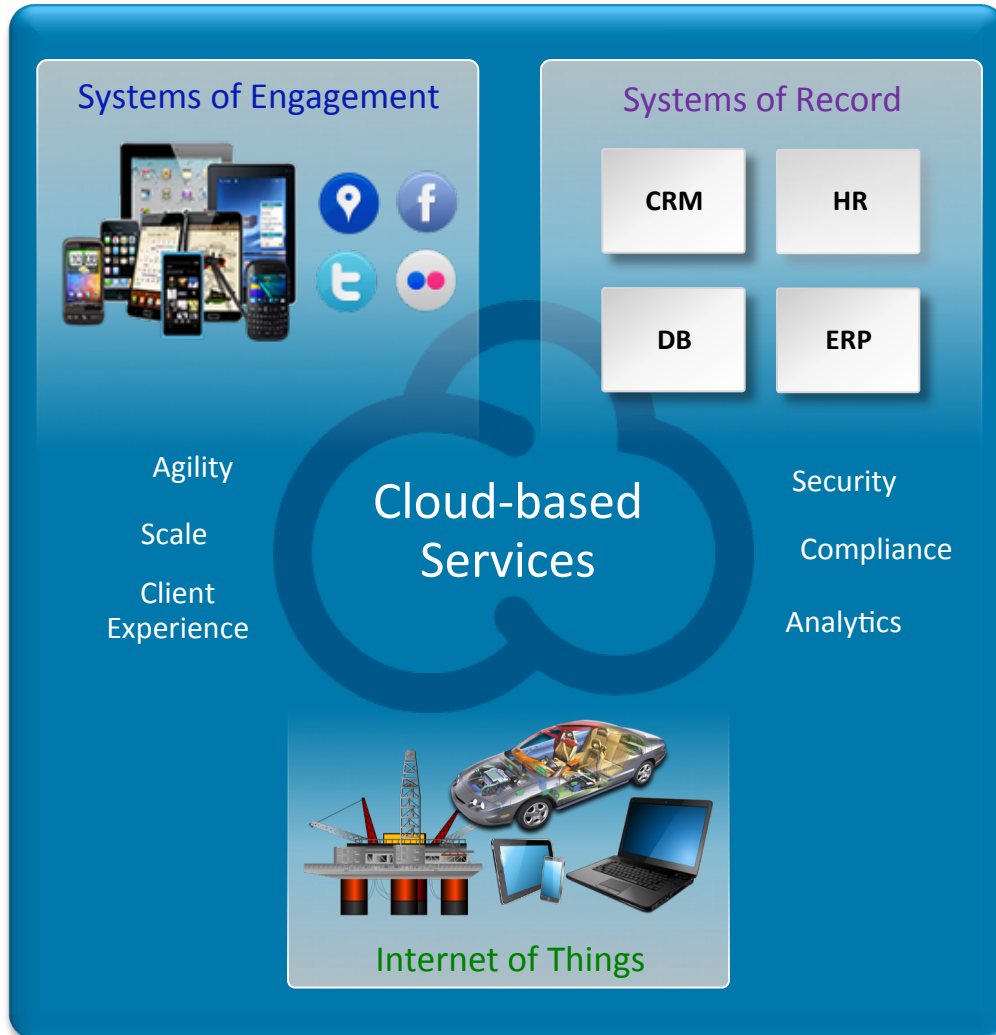
Scott Rich
Distinguished Engineer,
IBM Rational
srich@ch.ibm.com

Java in the Cloud, Today and Tomorrow

- Changing times

- Changing tools – the rise of PaaS

- Java with Eclipse, Liberty, Cloud Foundry and BlueMix

- Cloud-based tools for Cloud apps?

Ushering the New Era with Systems of Interaction

Systems of Engagement

Systems of Record

CRM   HR

DB   ERP

Agility

Scale

Client Experience

Cloud-based Services

Security

Compliance

Analytics

Internet of Things

Global Developer Population 2013

18M+ worldwide

North America 4M+

EMEA 6.4M

APAC 6M+

Latin America 1.7M

8M
Developing mobile Apps

4M+
Developing within Cloud

7M
Developing using node.js
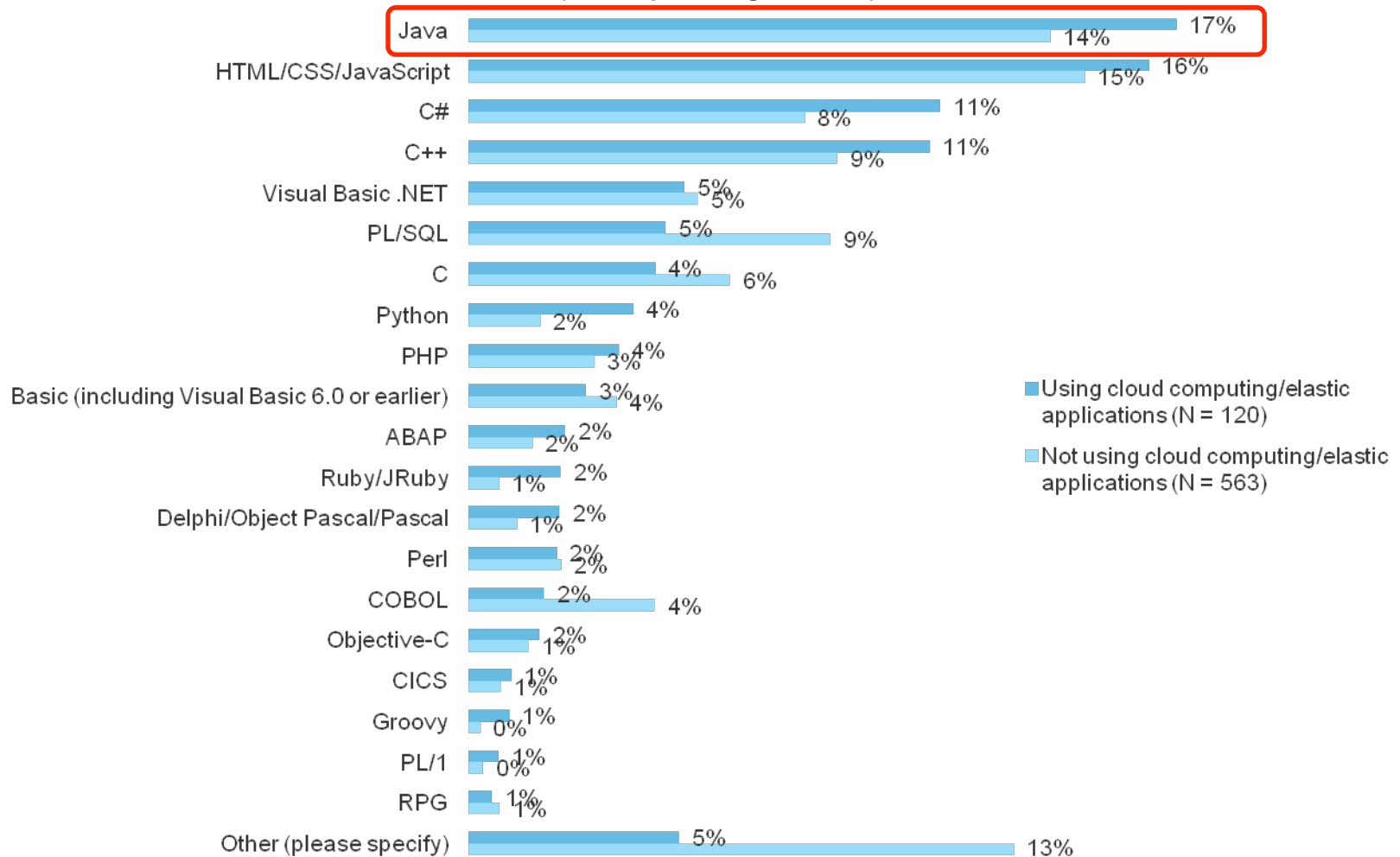
12M+
in at least one developer program

14M
Spend time with Agile (practices)

Growth Estimates:
- Estimated Global Dev Population by 2017: 23M+
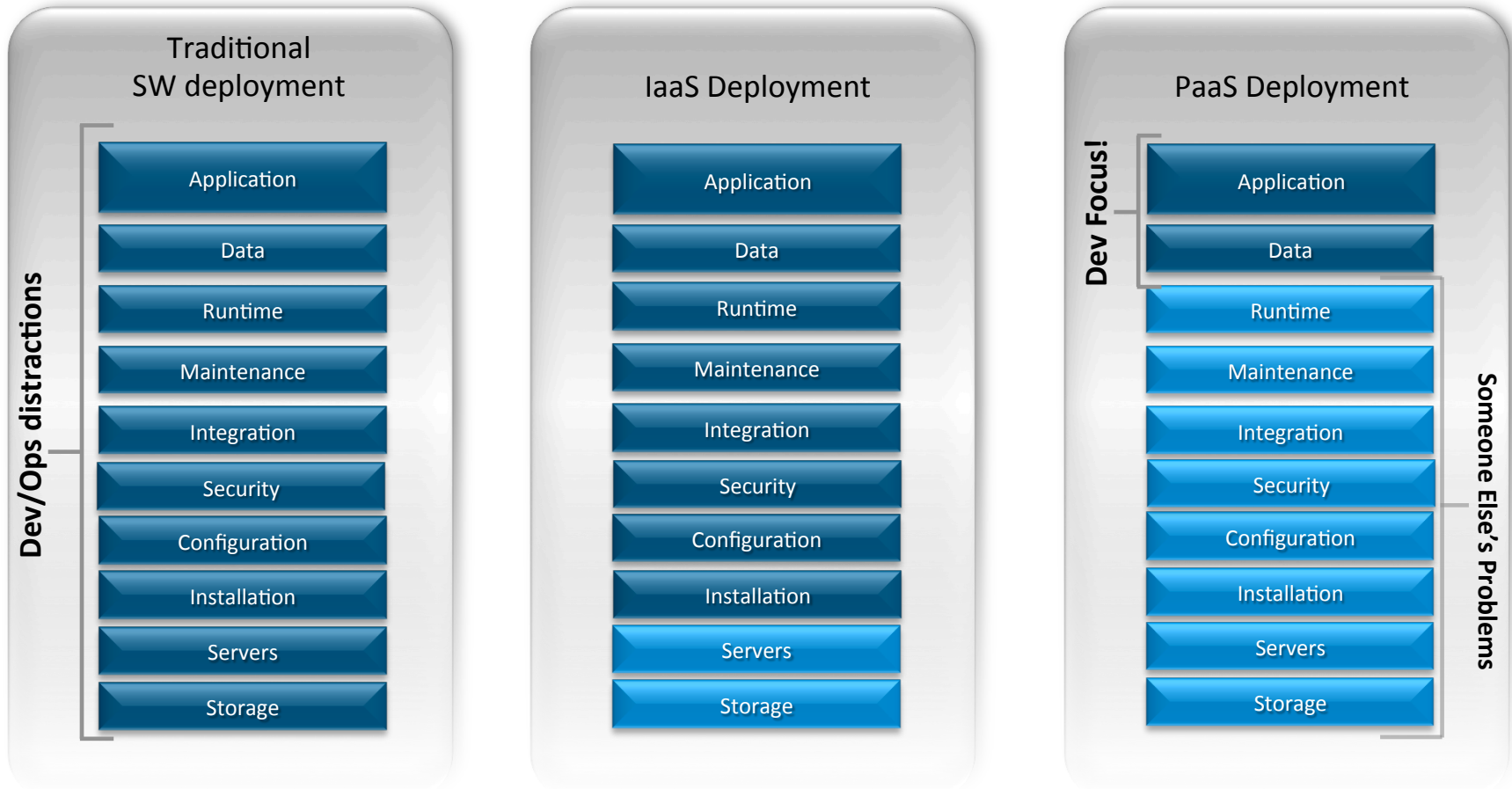- Estimated Mobile/Cloud Dev: over 70% of all new dev

Cloud developers use traditional enterprise languages

"How do you allocate the time you spend writing code across the following programming languages?"
(Enter a percentage for each)

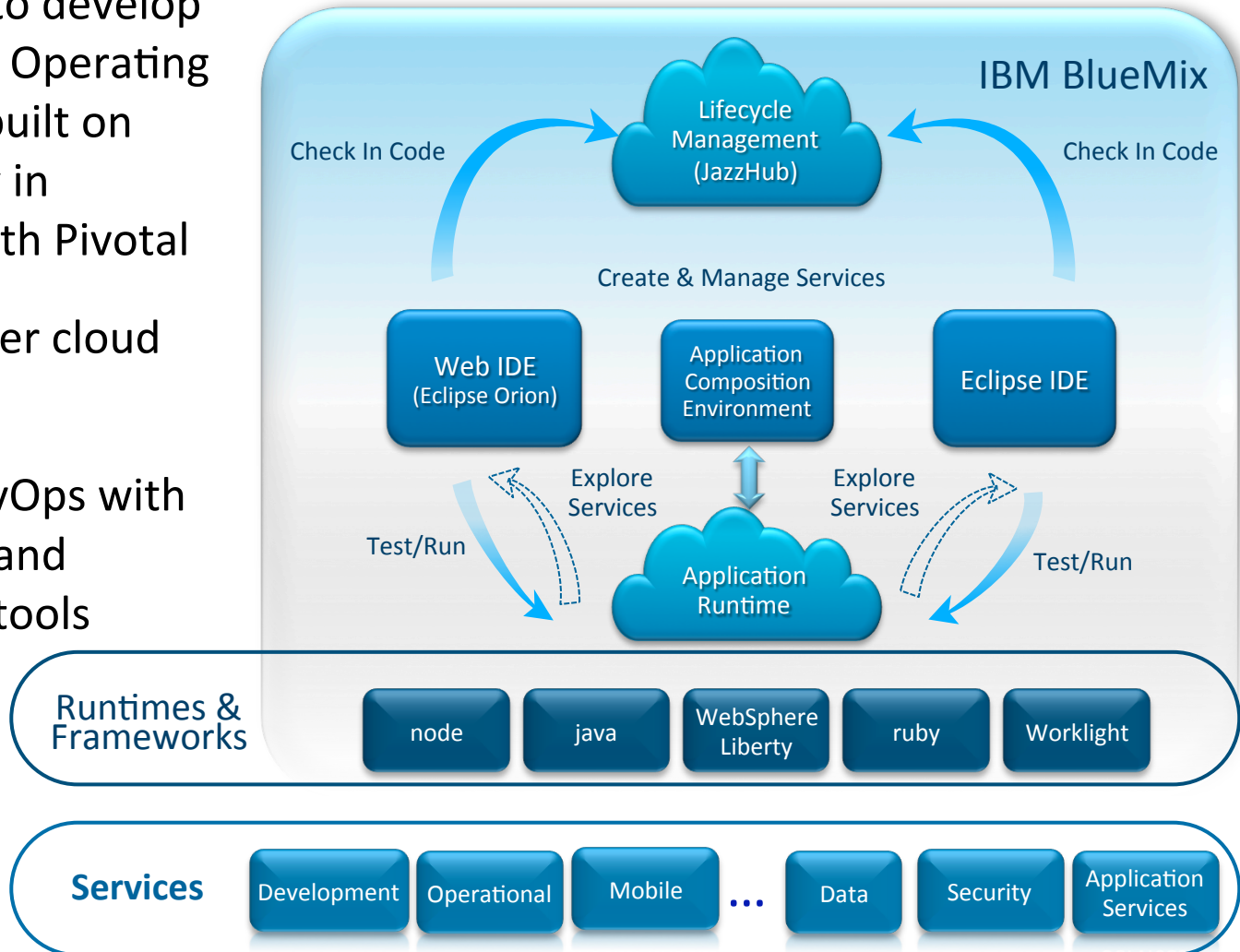| Language	Using cloud computing/elastic applications (N = 120)	Not using cloud computing/elastic applications (N = 563)
Java	17%	14%
HTML/CSS/JavaScript	16%	15%
C#	11%	8%
C++	11%	9%
Visual Basic .NET	5%	5%
PL/SQL	5%	9%
C	4%	6%
Python	4%	2%
PHP	4%	3%
Basic (including Visual Basic 6.0 or earlier)	4%	3%
ABAP	2%	2%
Ruby/JRuby	2%	1%
Delphi/Object Pascal/Pascal	2%	1%
Perl	2%	2%
COBOL	4%	2%
Objective-C	2%	1%
CICS	1%	1%
Groovy	1%	0%
PL/1	1%	0%
RPG	1%	1%
Other (please specify)	5%	13%

Base: North American and European enterprise software developers; Source: Forrsights Developer Survey, Q1 2013

"IaaS", "PaaS", what's the difference?

Traditional SW deployment	IaaS Deployment	PaaS Deployment
Application	Application	Application
Data	Data	Data
Runtime	Runtime	Runtime
Maintenance	Maintenance	Maintenance
Integration	Integration	Integration
Security	Security	Security
Configuration	Configuration	Configuration
Installation	Installation	Installation
Servers	Servers	Servers
Storage	Storage	Storage

Dev/Ops distractions

Dev Focus!

Someone Else's Problems

Introducing IBM BlueMix

- IBM initiative to develop an open Cloud Operating Environment built on Cloud Foundry in partnership with Pivotal

- IBM and partner cloud services

- Integrated DevOps with both Browser and Eclipse-based tools

Demo – developing Java for the Cloud with Eclipse

Demo setup

- Browser preference set for external

- Stop all servers and restart Eclipse

- Connect the VPN!!!

- Local servers stopped

- Connect to BlueMix and Pivotal
  – Remove HelloMongo from BlueMix

- Open HelloServlet and MongoServlet

Java demo script

- What have we got here?
    - Eclipse Juno Java EE edition
    - IBM Web Dev tools
    - Pivotal CloudFoundry integration
    - IBM Liberty runtime and tools

- Just doing really simple Web dev, a JSP and a couple servlets
    - Quick tour of HelloCloud

- Developing locally with Liberty
    - Servers list – debug "Liberty at localhost"
    - See the console, starts in 1 sec – "this is not your father's WebSphere"
    - click on the app URL, hit the breakpoint
    - Make a change and reload

- What if we want to do something a little more interesting, like use a Mongo DB?
    - Start the "Liberty with MongoDB" server
        - Don't have to stop, no port conflict

- Open the server definition – one page of XML
- Show MongoServlet
    - Using a local MongoDB
    - But it has a fall-back to allow Cloud Foundry to inject it – see getConnectionString()
- Run the app

- We've coded and debugged locally, now we want to deploy to the Cloud
    - Push to Pivotal WITHOUT service binding
    - Console shows push progress
    - Remote Systems-> files -> logs
    - Observe failure
    - Add service binding, update and restart
    - Observe login problem! ☹

- Finish with push to BlueMix, bind service

The Case for Developing in the Cloud

- More and more applications have moved to the Web, to mobile devices, and are being run in Cloud environments

- Most application lifecycle tools offer lifecycle capabilities in the Web
  - But Development tasks are largely still done on the desktop in environment like Eclipse or Visual Studio

- Could we offer this experience in the Web, and integrate it into the software lifecycle?

- This is what we're experimenting with at JazzHub, using the Eclipse Orion technology

Do developers really want to move to the Cloud?

- Progression of desktop to cloud-based tools

- Shift in developer knowledge sharing

- Deploying to Cloud-based PaaS, IaaS, etc.

- **How enjoyable was your last IDE setup experience ???**

What is Orion?

- Eclipse "platform" project initiated 2 years ago -- **Tools for the web, on the web**

- An evolution in tooling leveraging the impressive recent improvements in browser capabilities and availability of cloud services

- Embracing the shift to web-based application delivery and enabling cross site collaboration of "pluggable" services

- A community that benefits from compatibility between a common set of components and services to create richer cloud based tools than they might otherwise by themselves

Demo – developing in the browser at JazzHub

Demo setup

- Login to JazzHub

- Navigate to
https://hub.jazz.net/code/edit/edit.html#/code/file/srich/srich%20|%20Sentiment%20Analysis%20prototype/SimpleSentimentAnalysis/app.js

- Cf target https://api.bluemix.net

- Cf login

- Cf info

- Cf push

JazzHub demo script

- We're building our app for the Cloud, what if we could do all our coding there?

- This is our Web-based IDE, built on Eclipse Orion, hosted at JazzHub
  - Provides an extensible, high performance source code editor
  - Currently, most capable with the Web triple of HTML, CSS, JavaScript
  - Syntax highlighting, code checking with JSLint
  - Rich key bindings, a subset of the Eclipse keys

- Here, we're editing a JavaScript file as part of a Node.js app
  - We have basic syntax highlighting and content assist (not bad for an untyped language)

- Drop to the shell
  - Here's where we see the real power of the IDE
  - It's not just a widget
  - We've got a real file system behind this
    - Help
    - Ls
  - Because we've got real files we can integrate existing tools, like the CF command
    - Help cf
    - Cf info
  - I can push from here, but the real party trick is to Push from the IDE
  - Back to the editor
  - Push to BlueMix
    - Test
    - Go to the management console
    - This is one of the value add features we're building on top of CF, plus services

What could we do for Java developers with a PaaS at our disposal???

- Exploring server-side Java tool chain for Orion
  - There was a demo of a headless "JDT in the Cloud" at EclipseCon
  - Talking to potential collaborators

- Today, we're incredibly inefficient, re-indexing and compiling the same streams/branches on all our desktops
  - Better: stream-aware sharing of Java indices, maybe compiled JARs
  - We're exploring a flavor of this for Node.js modules already

- Server-side sharing of target-platforms, setup of dev environments(!!!)

- What can we do beyond that?
  - Continuous Integration and Continuous Deployment integration (PaaS-optimized)
  - Query-based semantic search: references, declarations, etc
  - Cross-workspace search
  - Code analytics?
    - Churn, Hotspots
    - Quality predictions based on developer behavior ("Big Coder"?)

Thanks! If you want to learn more…

- Try out Orion at JazzHub (https://hub.jazz.net) or OrionHub (http://orionhub.org/)

- Learn more about BlueMix at http://www.ibm.com/software/ebusiness/jstart/bluemix/index.html

- Learn more about Liberty at wasdev.net

- Get WDT, Liberty, and the Cloud Foundry tools from the Eclipse Marketplace