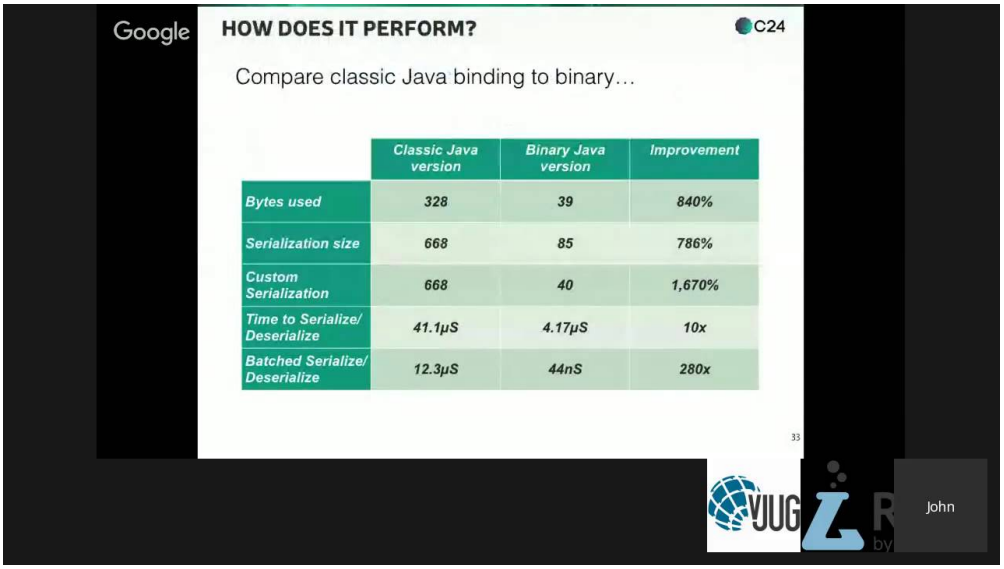
# Table of Contents

- Building Bootiful Applications with Spring Boot
- Java 8 Parallel Streams Workshop
- Project Lambda: Functional Prog. Constructs and Simpler Concurrency in Java SE 8
- WebSocket Applications using Java EE 7
- Comparing JVM Web Frameworks
- 55 New Features in Java SE 8
- How To Do Kick-Ass Software Development
- Getting started with Java EE 7
- Don’t be that guy! Developer Security Awareness
- Drive-by Contributions
- Design is a Process, not a Document

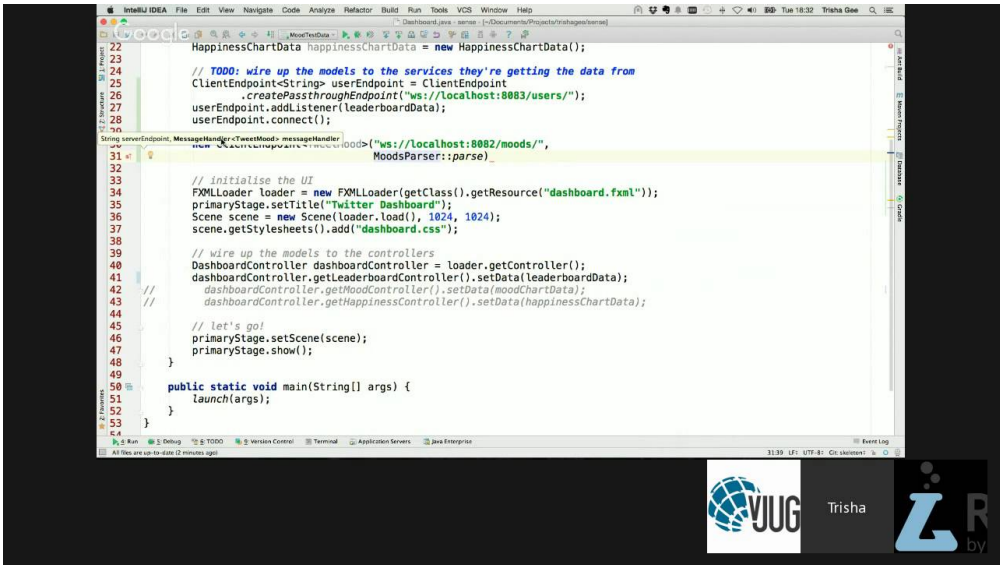
# Getting C/C++ performance out of Java




Today everything is connected, from the wearables to internet banking, every business is global and technology revolves around messaging. Data volumes have gone through the roof, “Big Data” is the buzzword of the era and we’ve been forced into distributed architectures and in-memory databases to handle the demand. We are now being held back by Java, its inefficient memory handling, and atrocious serialisation. By re-applying some of the lessons we learnt in the days of C/C++ we can significantly speed it up, and we’re not talking 50%, but 50 times.

http://virtualjug.com/?p=1878

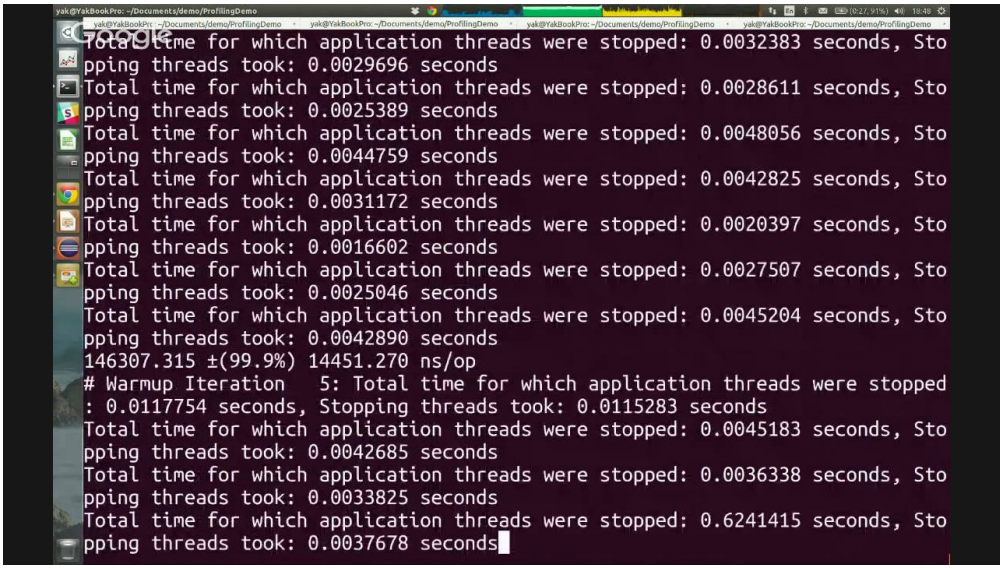
# Java 8 in Anger!




Java 9 is just around the corner, but many of us developers have yet to use Java 8 features in an application. This presentation moves beyond slide-sized examples of streams and lambdas, and to shows how to build a fully working end-to-end application using minimal external dependencies and the latest version of Java.


http://virtualjug.com/?p=1857

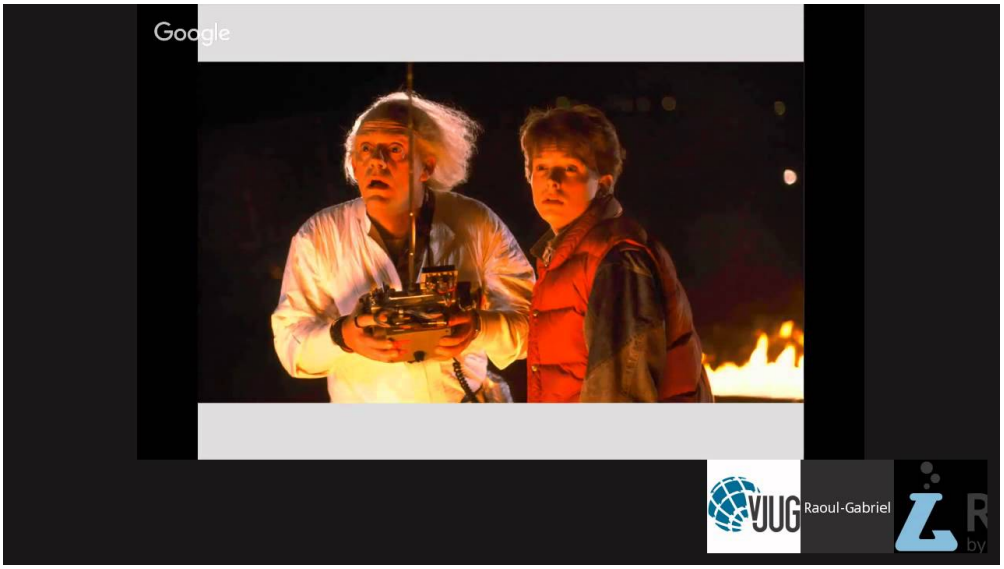
# Java Profiling from the Ground Up!




We will take a deep dive into the guts of Honest-Profiler, an unbiased sampling CPU profiler for the JVM, and into the JVM internals which enable it to work.


http://virtualjug.com/?p=1846

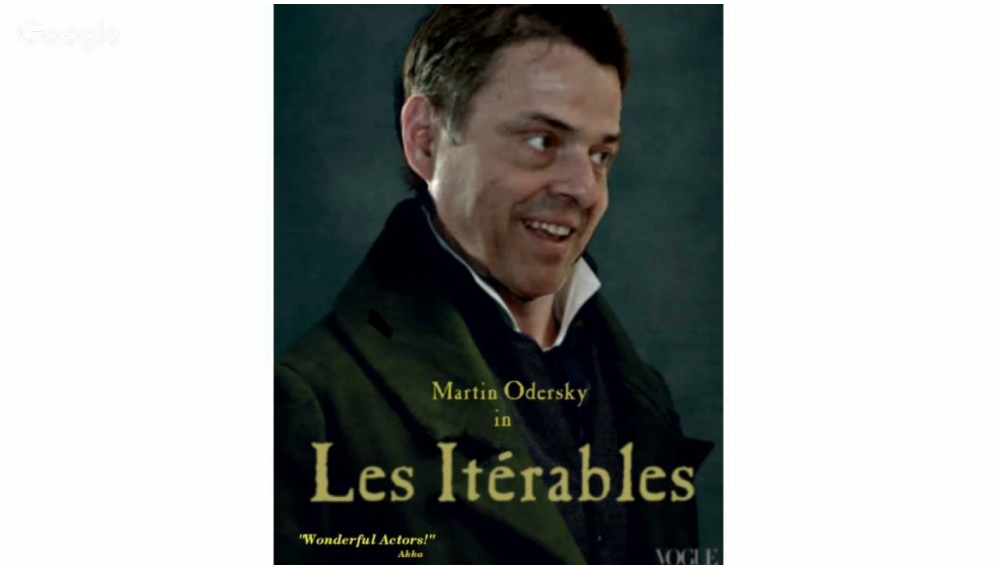
# Generics: Past, Present and Future




Generics are one of the most complex features of Java. They are often poorly understood and lead to confusing errors. Unfortunately, it won't get easier. Java 10, release planned for 2018, extends Generics. It's now time to understand generics or risk being left behind.

http://virtualjug.com/?p=1844

# Flavors of Java Concurrency




Writing concurrent code that is also correct is unbelievably hard. Naturally, humanity has developed a number of approaches to handle concurrency in the code, starting from basic threads that follow the hardware way to do concurrency to higher level primitives like fibers and work-stealing solutions. But which approach is the best for you?

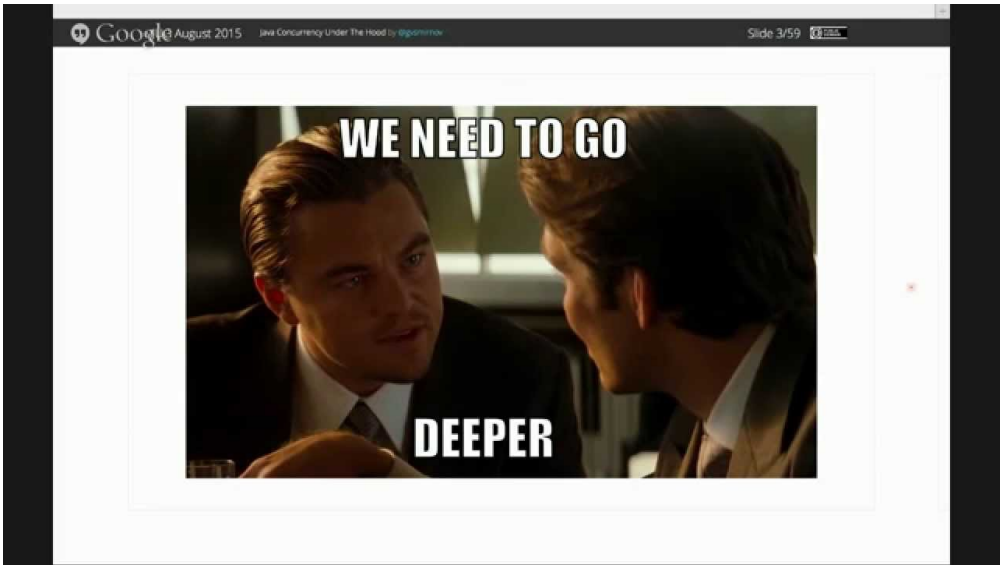

http://virtualjug.com/?p=1827

# Resilience is by design


Resilience; most developers understand what the word means, at least superficially, but way too many lack a deeper understanding of what it really means in the context of the system that they are working on now. I find it really sad to see, since understanding and managing failure is more important today than ever. Outages are incredibly costly—for many definitions of cost—and can sometimes take down whole businesses. In this talk we will explore the essence of resilience. What does it really mean? What is its mechanics and characterizing traits? How do other sciences and industries manage it? We will see that everything hints at the same conclusion; there is no “happy path”, failure is an option and resilience is by design. In this talk we will explore how.

http://virtualjug.com/?p=1807

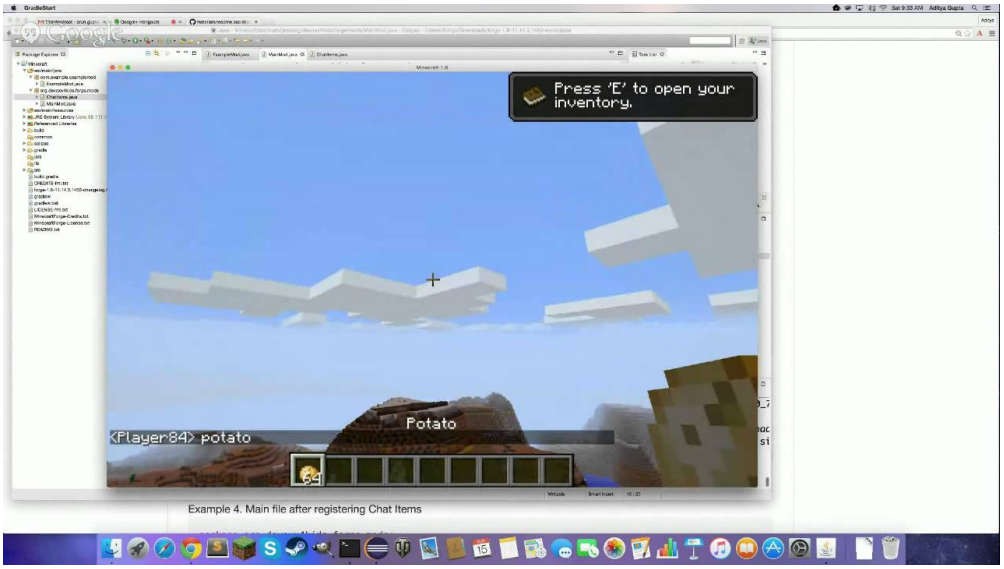
# Java Concurrency Under the Hood




In this age when parallelism matters, being able to write proper concurrent code is paramount. While Java hides lots of implementation details by its ‘Write Once, Run Anywhere’ motto, all abstractions will eventually leak. When they do, you will have to go deeper and see how that thing actually works.

http://virtualjug.com/?p=1799

# Getting Started with Minecraft Modding



Example 4. Main file after registering Chat Items

In this session, we'll show parents and kids how to get started building Minecraft mods with Minecraft Forge. We'll show you how to setup your computer with little fuss, as well as walk you through the process of creating your first mod. You'll also learn essential Java programming skills. If you're a kid searching for new ways to have fun with the game, or a parent looking to nurture your kids' creativity through code, you won't want to miss this exciting, hands-on tutorial.

VJUG

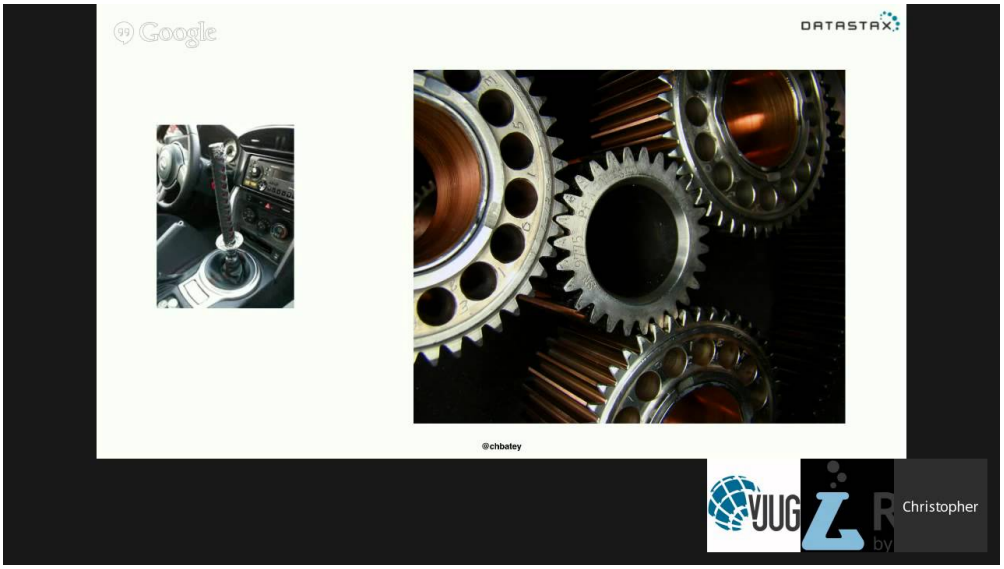
http://virtualjug.com/?p=1794

# Value in Relationships – How Graphs make databases fun again


Looking at the world around us – society, social, science, economy and tech we can’t see any isolated pieces of information. Instead everything is densely connected and a lot of the valuable information lives in the relationships between your entities.In the past and present databases always had a hard time to manage highly connected and semi-structured information in an efficient manner.

http://virtualjug.com/?p=1745

# So why would I use a distributed database like Apache Cassandra?

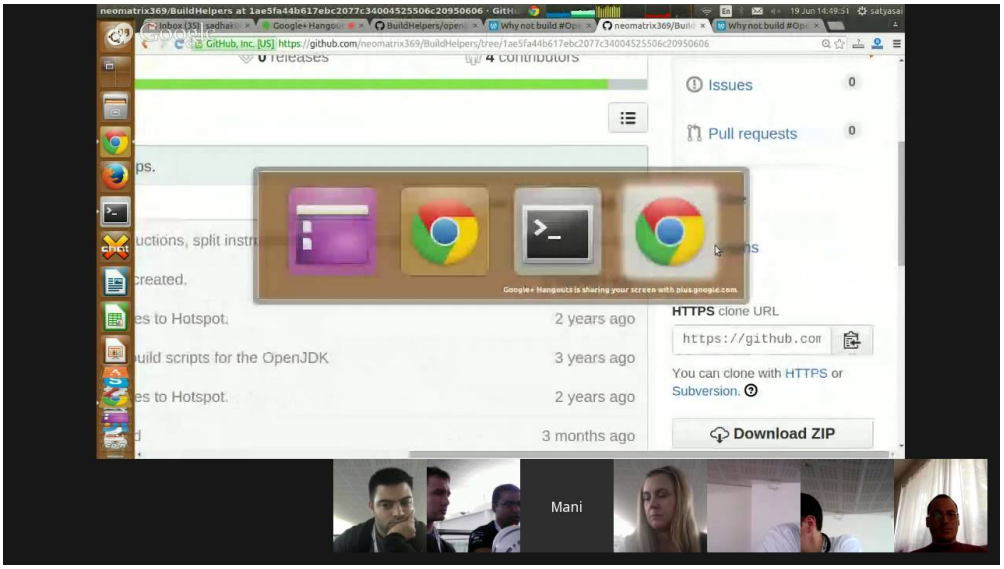



A new “database” seems to appear every other week. Most of them are “NoSQL” so they must be cool. All these new tools make it really hard for developers to cut through the fluff and know which type of data store to use and why.

VJUG

http://virtualjug.com/?p=1716

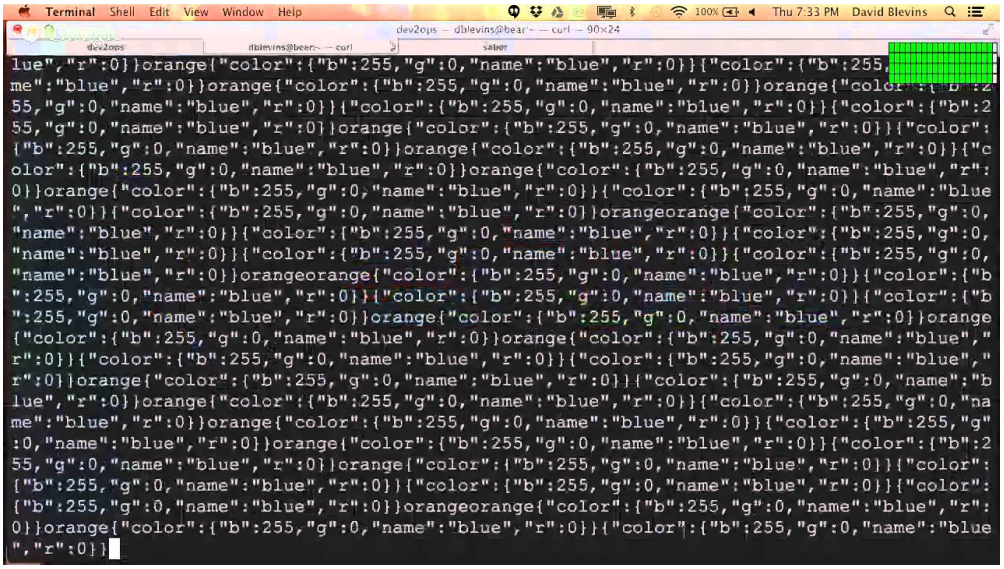
# Live From Devoxx UK Hackergarten: From vJUG virtuality to Devoxx UK real




One dev: Did you know what happened at a recent Java conference few months ago ?

http://virtualjug.com/?p=1714

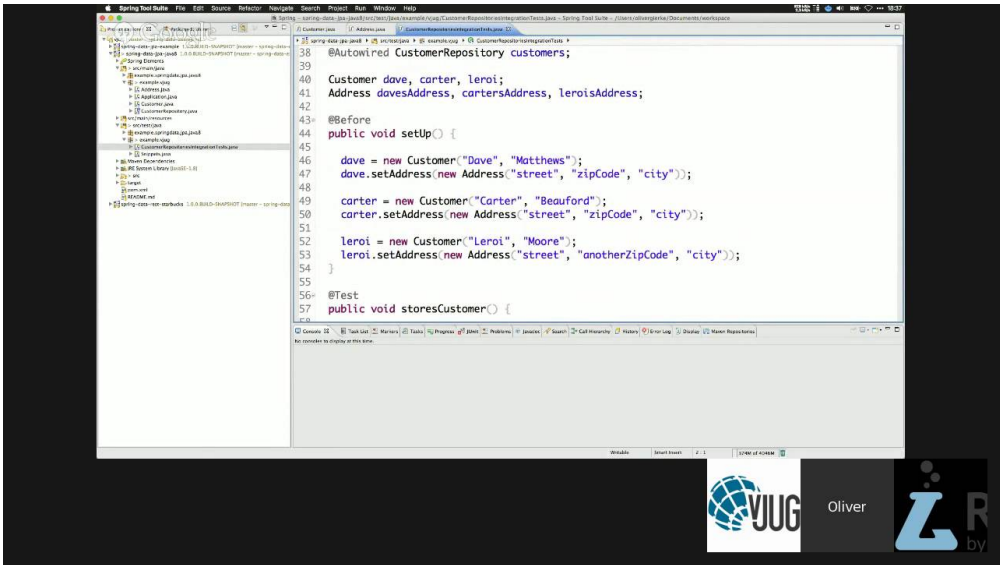
# Live From Devoxx UK: Apache TomEE from Dev to Ops




Apache TomEE is the Java EE distribution of Apache Tomcat. This live vJUG session goes beyond the basics and explores some fun features both TomEE-specific and JavaEE-portable for supercharging your application development, runtime and maintenance. Have a huge pile of DAOs? Use TomEE's abstract bean concept. Need to configure your application for many different environments? CDI and portable-extensions to the rescue. Want to create secured microservice distributions without any fuss? Nothing beats the TomEE Maven Plugin. Looking for a way to get detailed stats from your code? Hello annotation-driven monitoring support. Ever wish you could make your own management API? Check out the portable SSH Connector.The perfect session for any TomEE or Java EE enthusiast looking for cool toys for both developer and operations bliss.


http://virtualjug.com/?p=1711

# State of the art data access with Spring Data




Even with the invention of JPA, implementing data-access layers in Java has been a tedious job for developers, often resulting in a lot of boilerplate code. Spring Data is an umbrella project that provides a convenient and consistent interface-based programming model to implement repositories. It can be used on top of JPA as well as NoSQL stores like MongoDB and Neo4j.

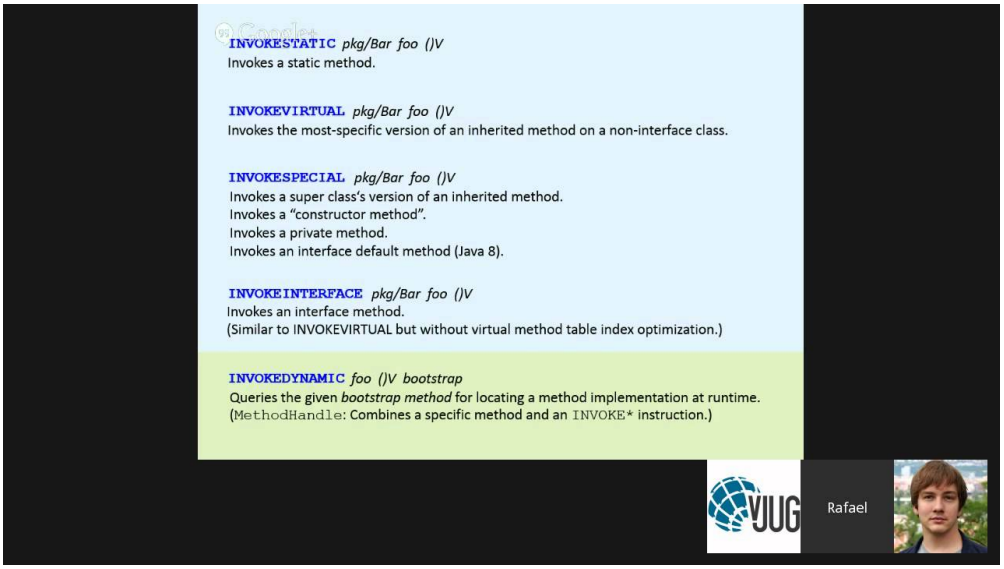

http://virtualjug.com/?p=1709

# Gradle: hot or not?


Maven has been the preferred build tool of many Java based projects for years however times are a-changing, there’s a new build tool in town and it promises to speed up build times, deliver build consistency, easier CI setup, extensibility and more. This tool is Gradle. Prominent open source projects have switched to Gradle already; organizations around the world are evaluating it too or made the switch already. So what makes Gradle tick? Join us to figure out the details! After all, it’s not a “should I change to Gradle” question, rather “_when_ should I change to Gradle”.

http://virtualjug.com/?p=1704

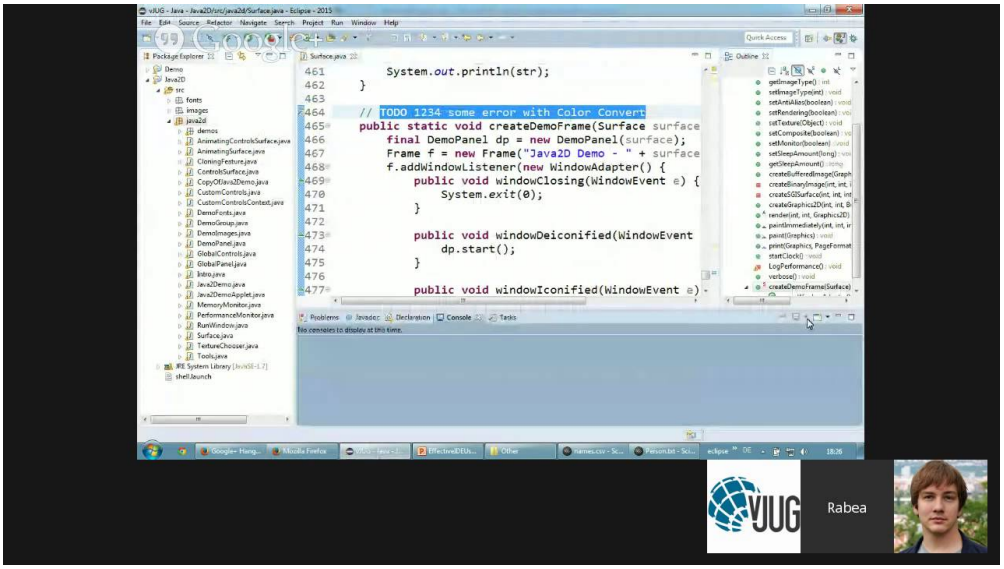
# Java byte code in practice




At first glance, Java byte code can appear to be some low level magic that is both hard to understand and effectively irrelevant to application developers. However, neither is true. With only little practice, Java byte code becomes easy to read and can give true insights into the functioning of a Java program. In this talk, we will cast light on compiled Java code and its interplay with the Java virtual machine. In the process, we will look into the evolution of byte code over the recent major releases with features such as dynamic method invocation which is the basis to Java 8 lambda expressions. Finally, we will learn about tools for the run time generation of Java classes and how these tools are used to build modern frameworks and libraries. Among those tools, I present Byte Buddy, an open source tool of my own efforts and an attempt to considerably simplify run time code generation in Java. (http://bytebuddy.net)

http://virtualjug.com/?p=1673

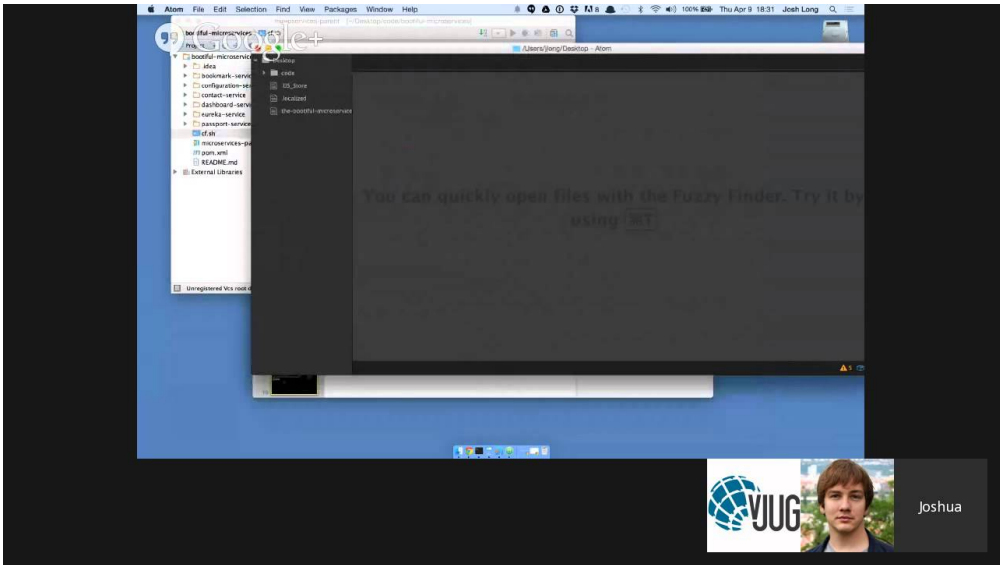
# Effective IDE Usage




You don’t want your IDE to propose java.awt.List as import when you need java.util.List? This talk will show you how to get rid of the proposal and how to use your IDE effectively to concentrate on your work.

http://virtualjug.com/?p=1658

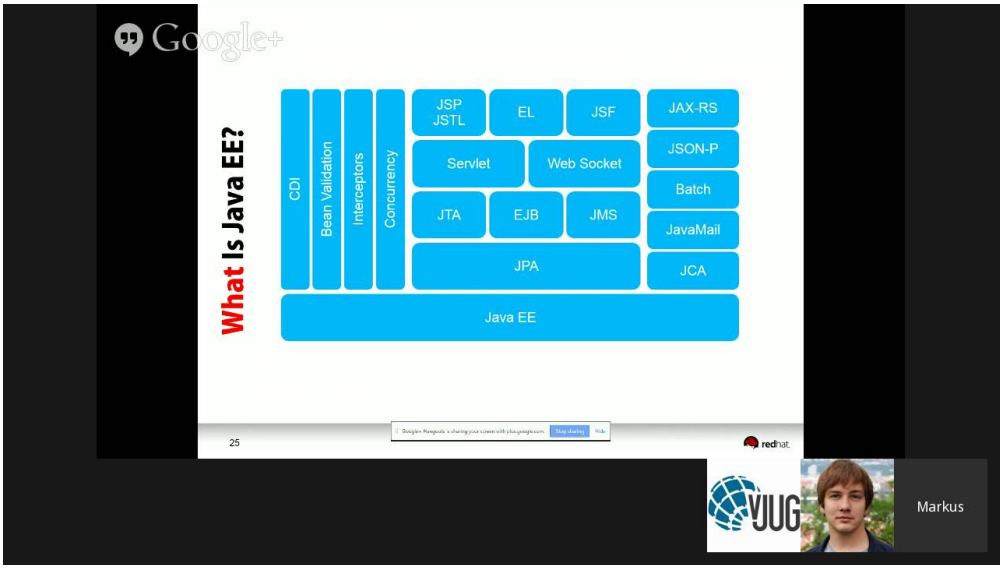
# Building “Bootiful” Microservices with Spring Cloud




We get it already! Microservices help you build smaller, singly-focused services, quicker. They scale out. They’re more agile because individual teams can deliver them at their own pace. They work well in the cloud because they’re smaller, and benefit from elastic, horizontal scaling. But what about the complexity? There’s a cost associated with adding services and coordinating the interactions between them. In this talk, we’ll look at Spring Cloud, which builds atop Spring Boot and the Netflix OSS stack, and see how it lets you easily integrate service-discovery, security, reliability patterns like the circuit breaker, and centralized and journaled property configuration (and more) to build resilient microservices that scale.

http://virtualjug.com/?p=1552

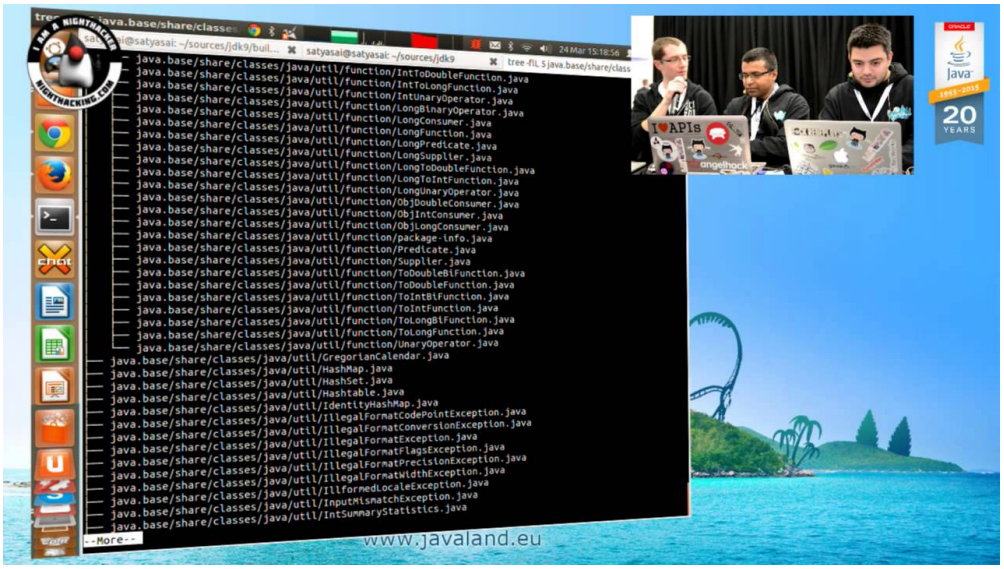
# Architecting Large Enterprise Java Projects




In the past I've been building component oriented applications with what I had at hand. Mostly driven by the features available in the Java EE standard to be "portable" and easy to use. Looking back this has been a perfect fit for many customers and applications. With an increasing demand for highly integrated applications which use already available services and processes from all over the place (departmental, central or even cloud services) this approach starts to feel more and more outdated. And this feel does not come from a technology perspective but from all the requirements around it. Having this in mind this post is the starting point of a series of how-to's and short tutorials which aim to showcase some more diverse ways of building (Java EE) applications that fit better into today's requirements and landscapes.

http://virtualjug.com/?p=1546

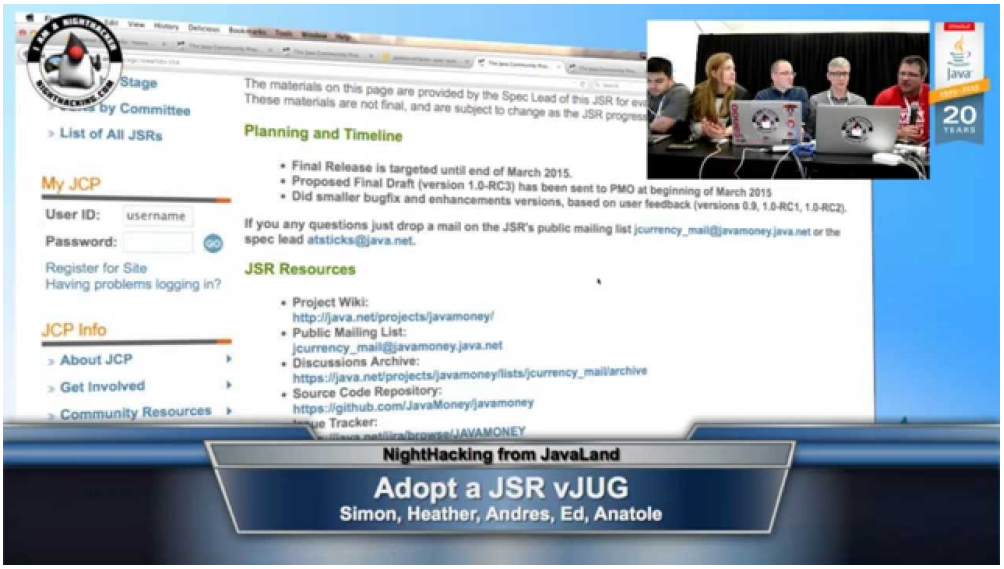
# JavaLand Session: How is Java/JVM built?




**Mr Webber developer** and **Ms Janet Java developer** are both developers who are interested in broadening their know-how of the Java platform. **Mr Webber developer** shares with **Ms Janet Java developer** conversations about Javaland, vJUG, Nighthacking and Adopt OpenJDK – a preview of their conversation.

http://virtualjug.com/?p=1533

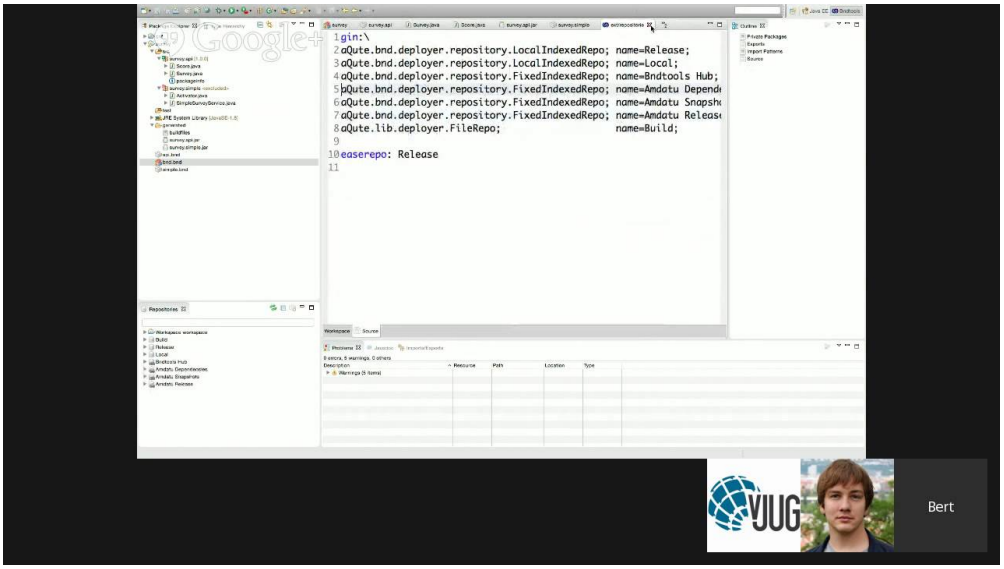
# JavaLand Session: What’s coming in Java.Next?




This session will take place live from the Javaland conference in Germany on the Nighthacking stage! Learn from **Heather VanCura** how you can take part in Java technology by Adopting a JSR. This session give a brief overview of the Adopt-a-JSR program and how to participate through the Virtual JUG. We will meet and discuss with three current JCP Spec Leads to find out how their JSRs could benefit from vJUG Adopt-a-JSR participation.

http://virtualjug.com/?p=1532

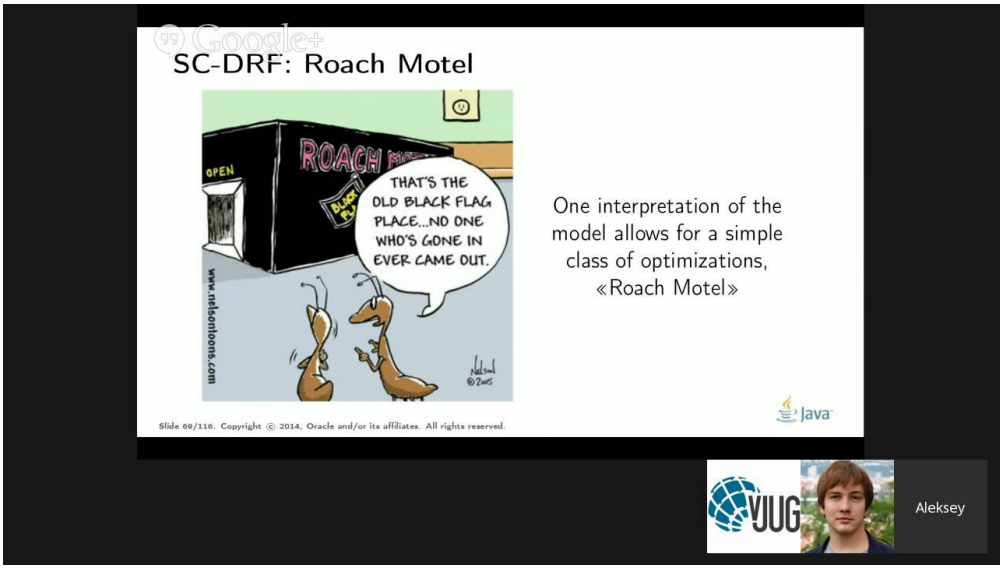
# Building Modular Java Applications in the Cloud




Modularity is an architectural theme that you'll hear about more and more. Being able to deal with change in a codebase is not something trivial and requires some serious thought. In this talk I will show you that it is actually pretty easy to achieve a modular architecture using OSGi, and the right set of tools. Of course everything will be demonstrated using live coding!

http://virtualjug.com/?p=1436

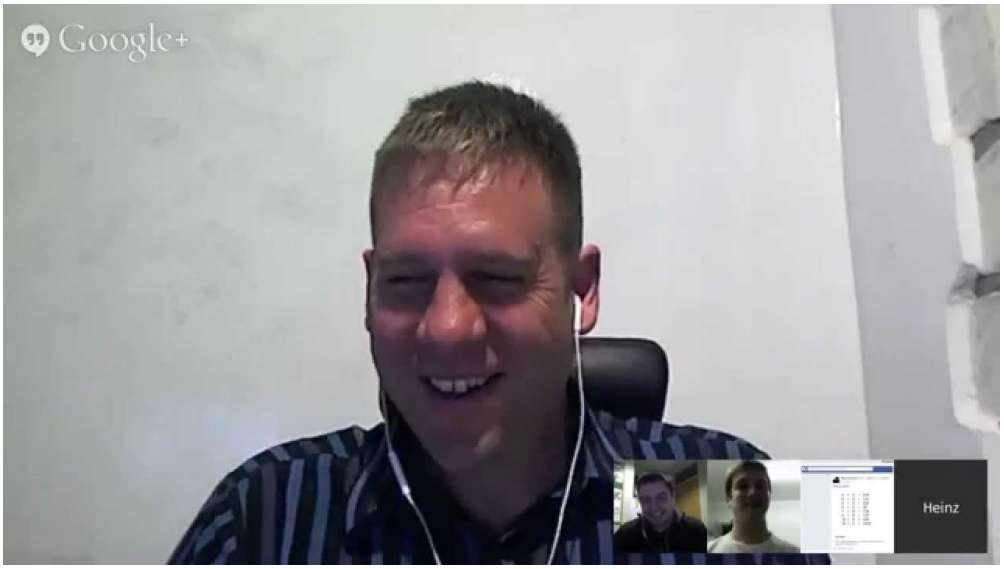
# Java Memory Model Pragmatics




The Java Memory Model is the most complicated part of Java spec that must be understood by at least library and runtime developers. Unfortunately, it is worded in such a way that it takes a few senior guys to decipher it for each other.

http://virtualjug.com/?p=1388

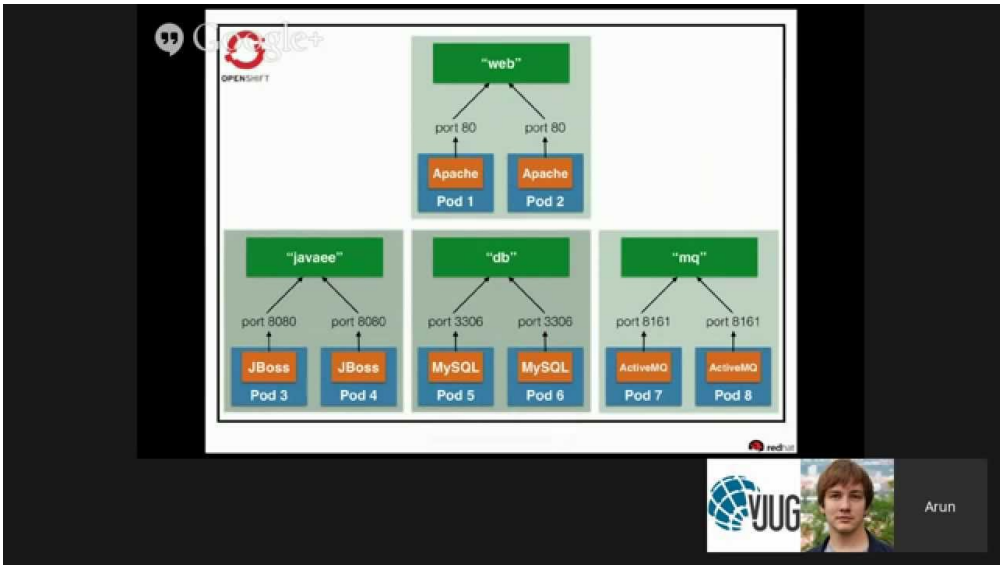
# The Live Reflection Madness




Heinz likes to compare reflection to opium. Not the perfume. The drug. In this live coding session, he will start by showing some of the powerful features available to us in Java.

http://virtualjug.com/?p=1347

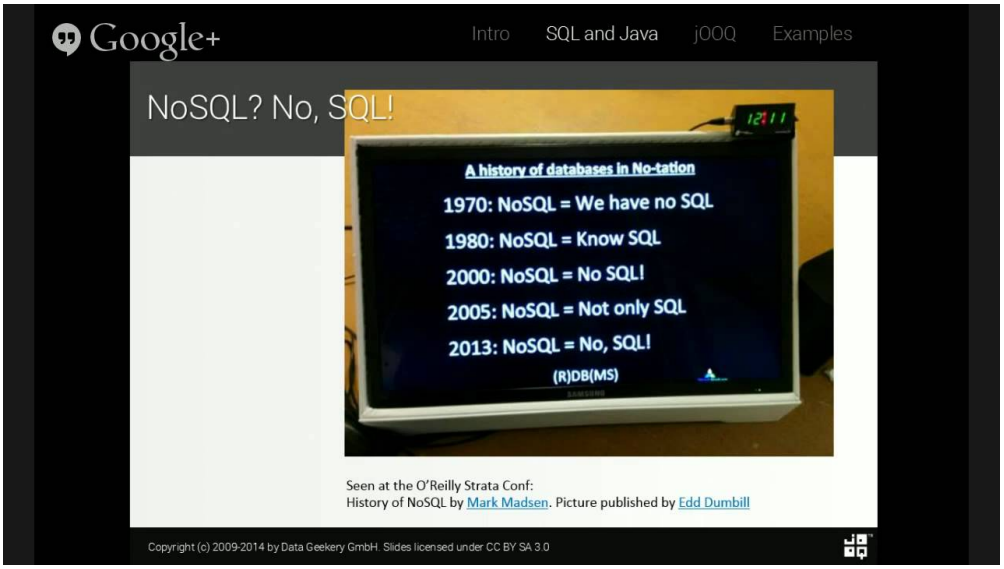
# Package your Java EE application using Docker and Kubernetes




Docker simplifies software delivery by making it easy to build and share images that contain your application's operating system. It packages your application and infrastructure together, managed as one component.

http://virtualjug.com/?p=1343

# jOOQ: Get Back in Control of Your SQL




SQL is a powerful and highly expressive language for queries against relational databases. SQL is established, standardised and hardly challenged by alternative querying languages.

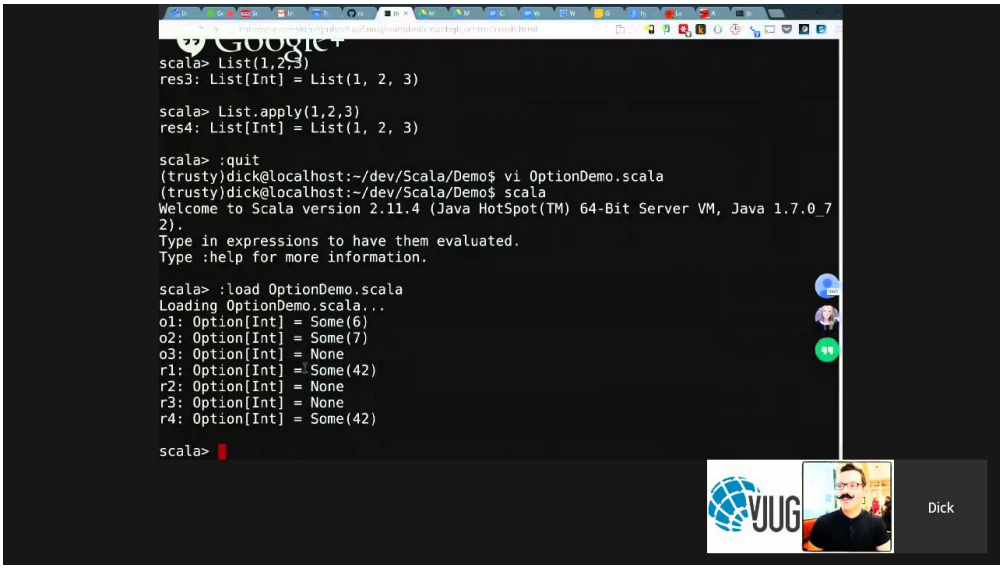

http://virtualjug.com/?p=1337

# Java and the Wave Glider, by James Gosling


IRC logs be be found here.

http://virtualjug.com/?p=1325

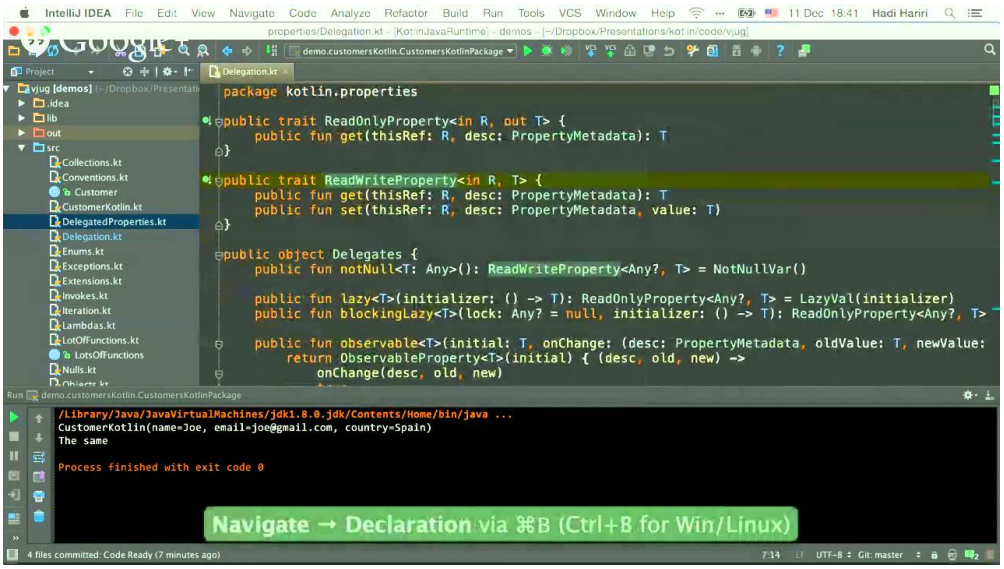
# Scala for Java Developers




– What are the major advantages/features Scala provides
– Why should someone move from Java to Scala
– What is the future direction of Scala


http://virtualjug.com/?p=1302

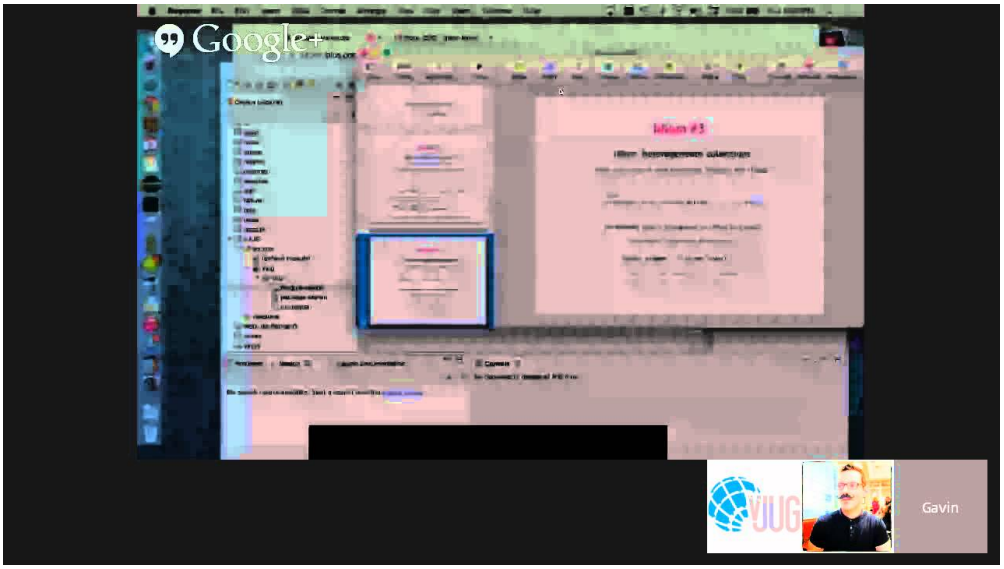
# Kotlin for Java Developers




– What are the major advantages/features Kotlin provides
– Why should someone move from Java to Kotlin
– What is the future direction of Kotlin


http://virtualjug.com/?p=1297

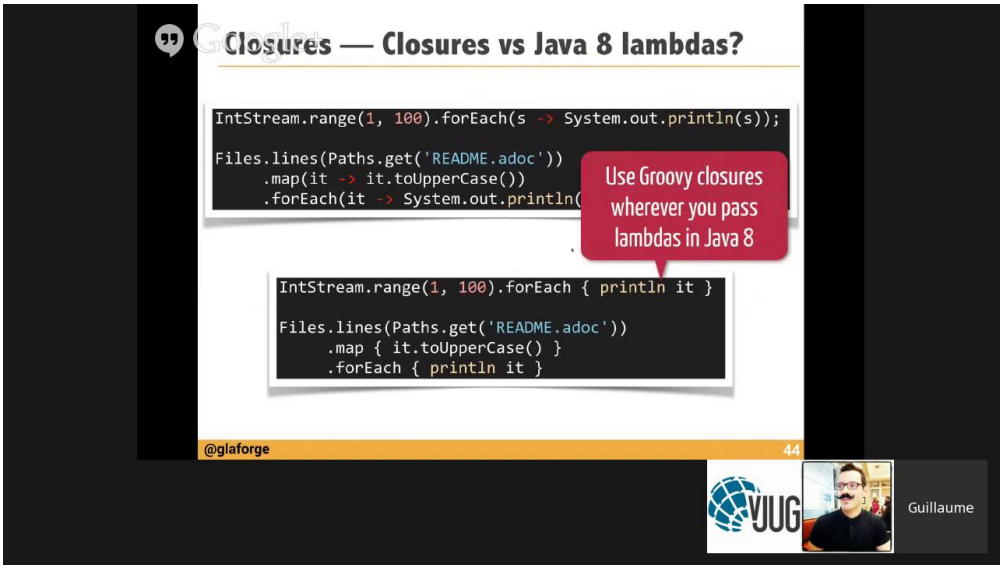
# Ceylon for Java Developers




– What are the major advantages/features Ceylon provides
– Why should someone move from Java to Ceylon
– What is the future direction of Ceylon

http://virtualjug.com/?p=1294

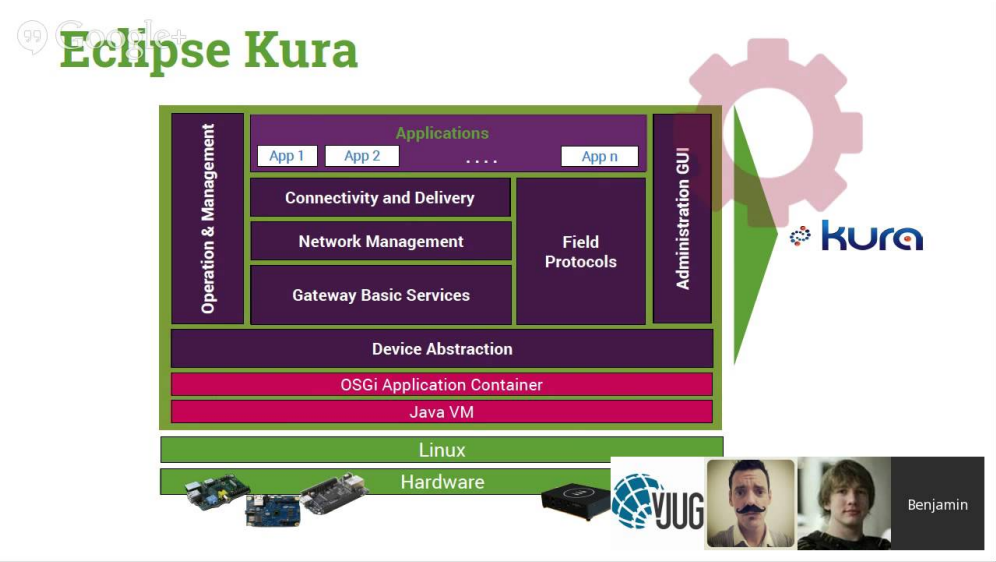
# Groovy for Java Developers




**Speaker: Guillaume Laforge**

http://virtualjug.com/?p=1288

# Building the Internet of Things with Java

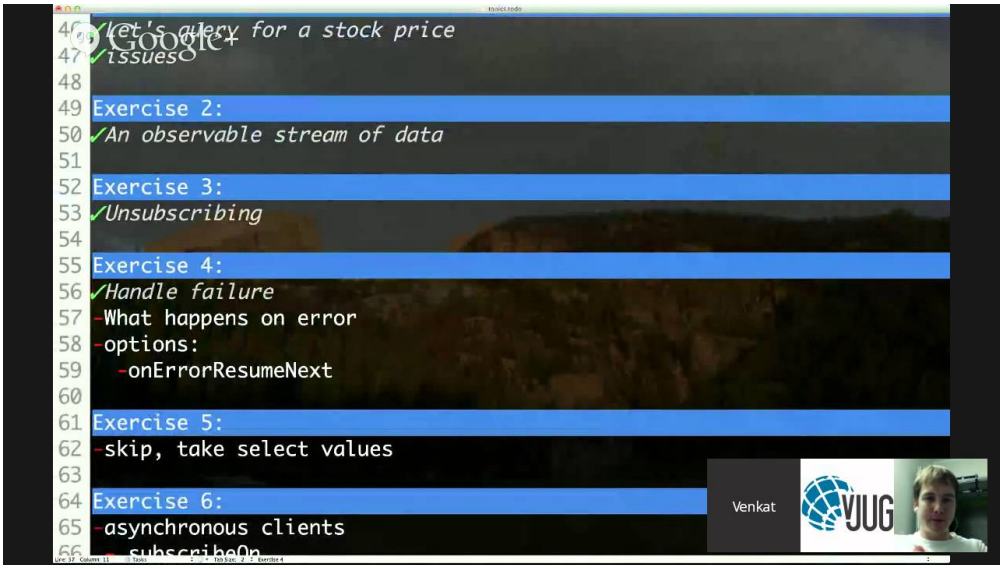



It may seem hard to get started with the Internet of Things (IoT) with so many technologies, protocols, hardware platforms, involved. In this session, Benjamin Cabé from the Eclipse Foundation will cover all you need to know

VJUG

http://virtualjug.com/?p=1195

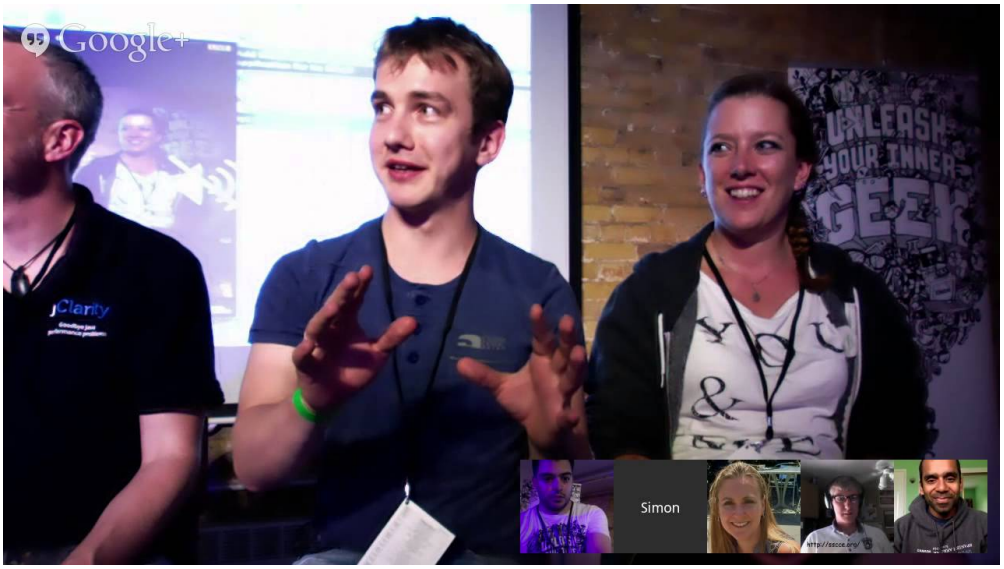
# Reactive Programming: Creating highly responsive applications




Reactive Programming in gaining a lot of attention recently, but what is it? It is a culmination of a lot of good ideas developed over the years, but brought together by the forces of recent developments

http://virtualjug.com/?p=1193

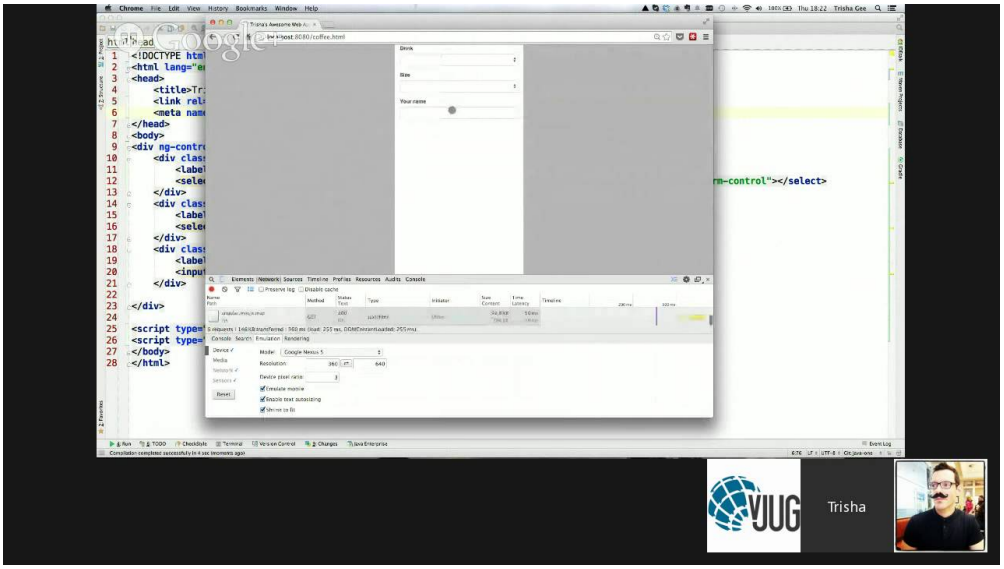
# Shaping Java’s future & vJUG party!




**Shaping Java’s future & vJUG party!**


http://virtualjug.com/?p=1191

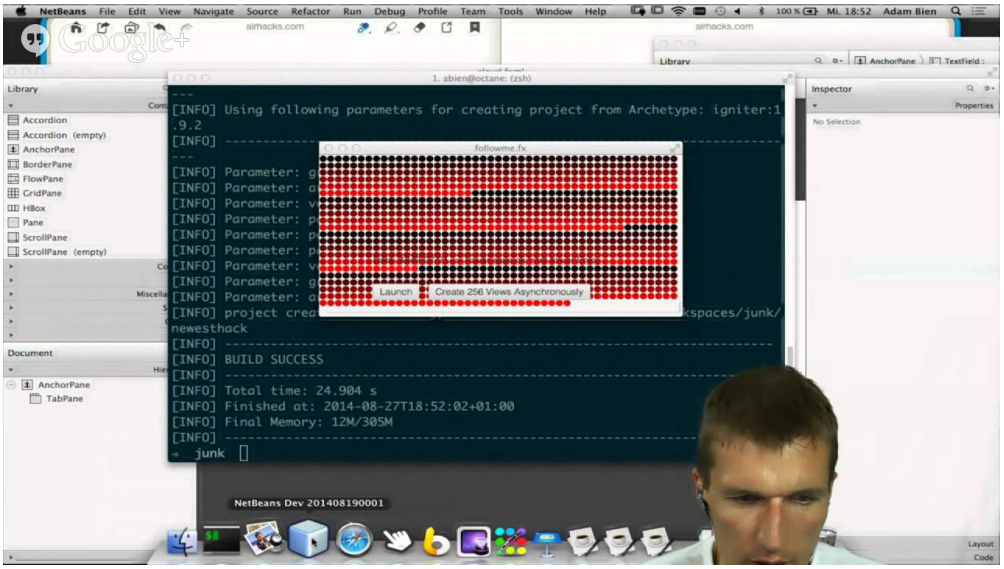
# HTML5, AngularJS, Groovy, Java and MongoDB all together – what could go wrong?




**Speaker: Trisha Gee**

http://virtualjug.com/?p=1188

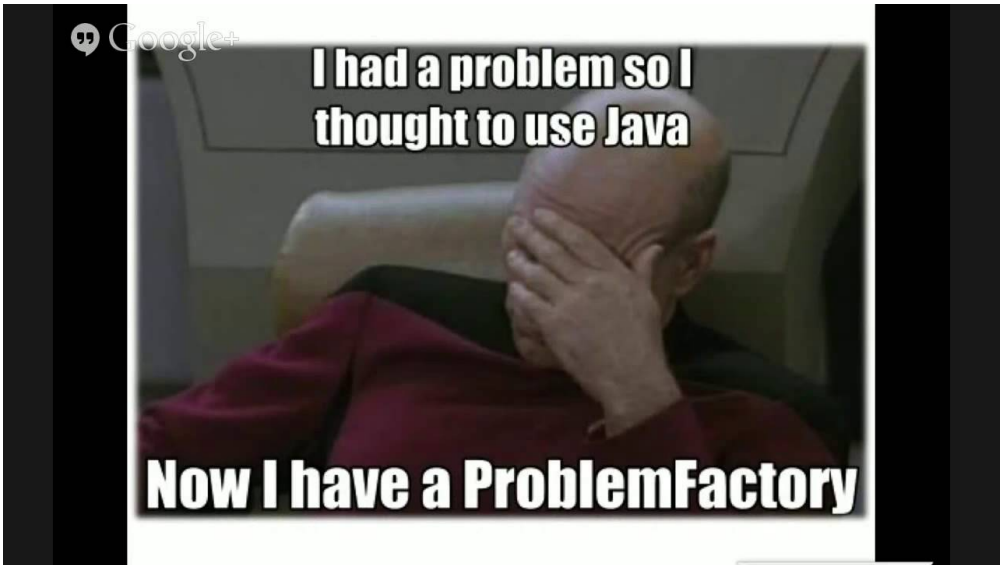
# Opinionated JavaFX 8




**Speaker: Adam Bien**

http://virtualjug.com/?p=1157

# 3 years of backend testing at Shazam [the stuff we got wrong]




**Speaker: Colin Vipurs**

VJUG

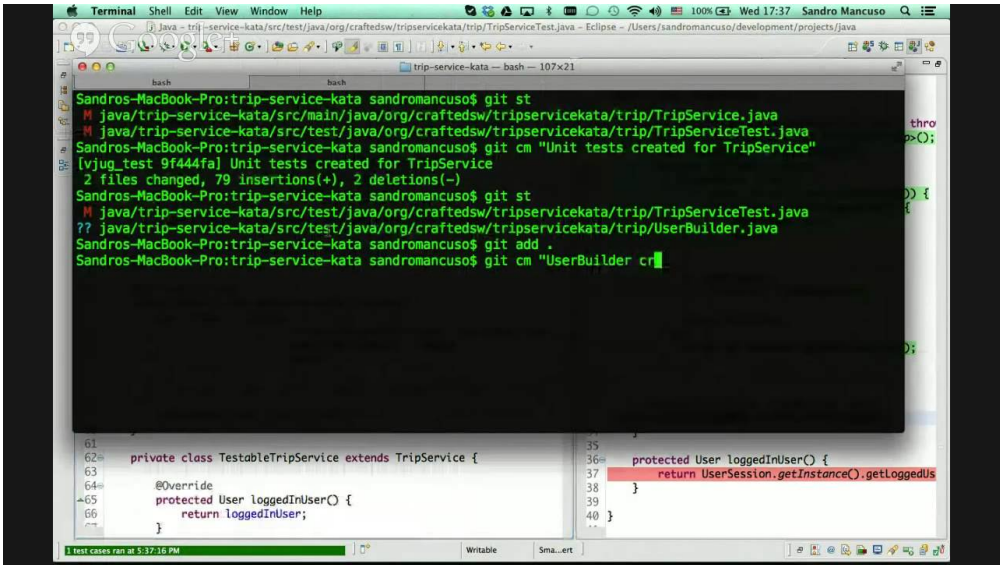
http://virtualjug.com/?p=1155

# Pragmatic Functional Refactoring with Java 8


**Speakers: Richard Warburton & Raoul-Gabriel Urma**


http://virtualjug.com/?p=1152

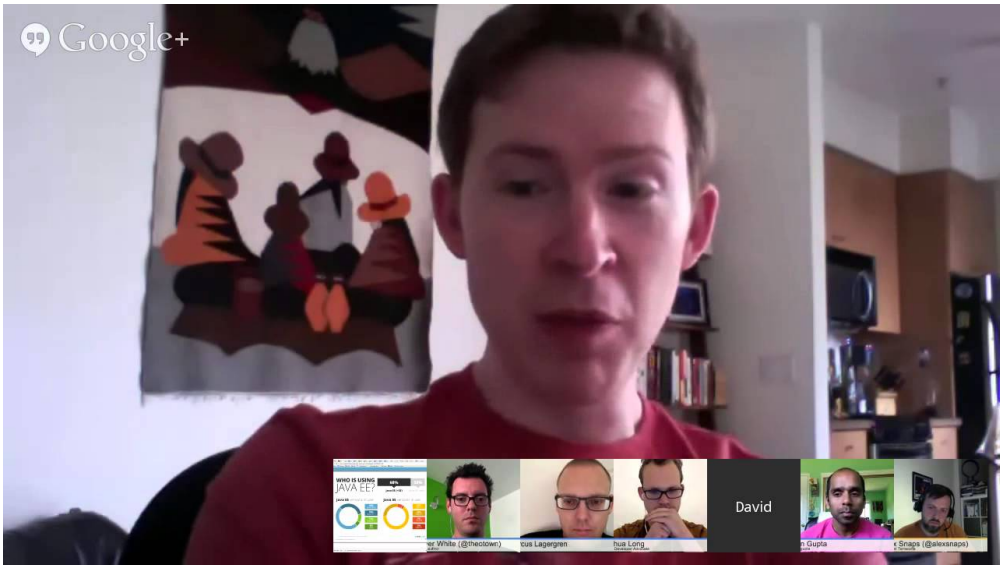
# Testing and Refactoring Legacy Code




**Speaker: Sandro Mancuso**

http://virtualjug.com/?p=993

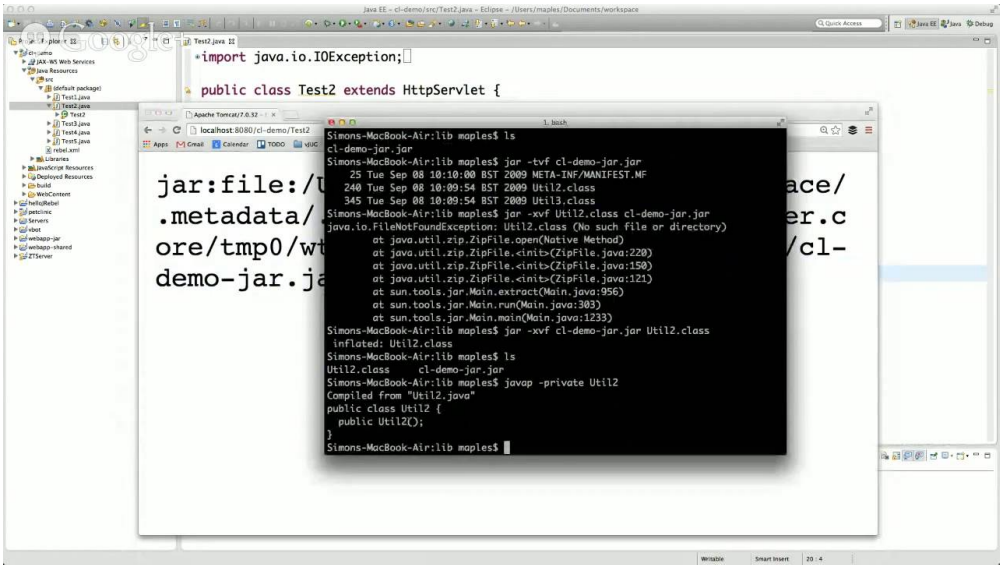
# vJUG panel: Review of 2164 Survey Responses on Java Tools and Technology




Speakers: Arun Gupta, Josh Long, Marcus Lagergren, Alex Snaps, David Blevins & Oliver White (Moderator). We look at the recent explosive publication of RebelLabs' "Java Tools and Technologies Landscape for 2014", a beautifully-designed, 56-page snapshot of what over 2000 Java developers from around the world are using in their daily development.

http://virtualjug.com/?p=938

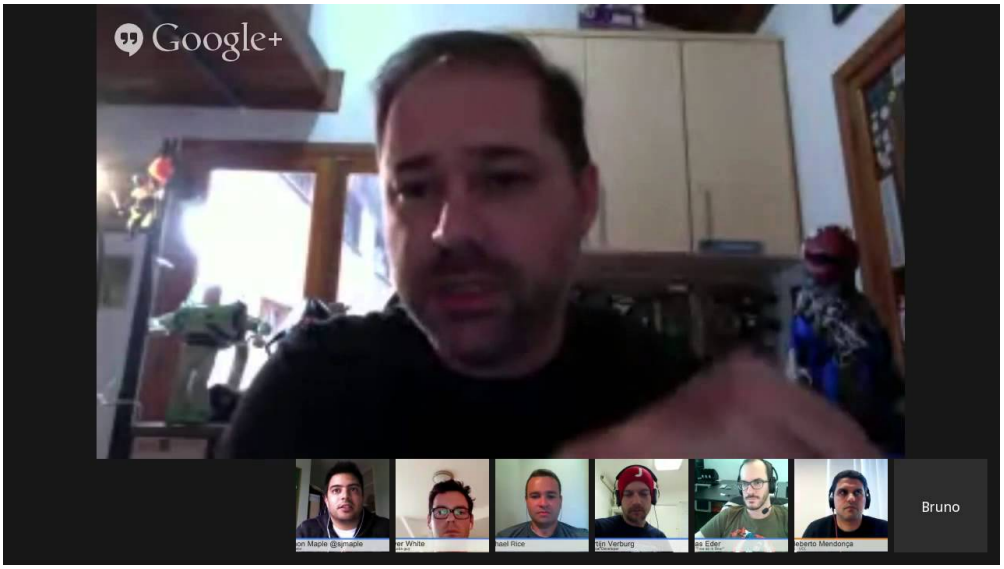
# Java Classloaders: The good, the bad and the WTF.




Speaker: Simon Maple.


http://virtualjug.com/?p=936

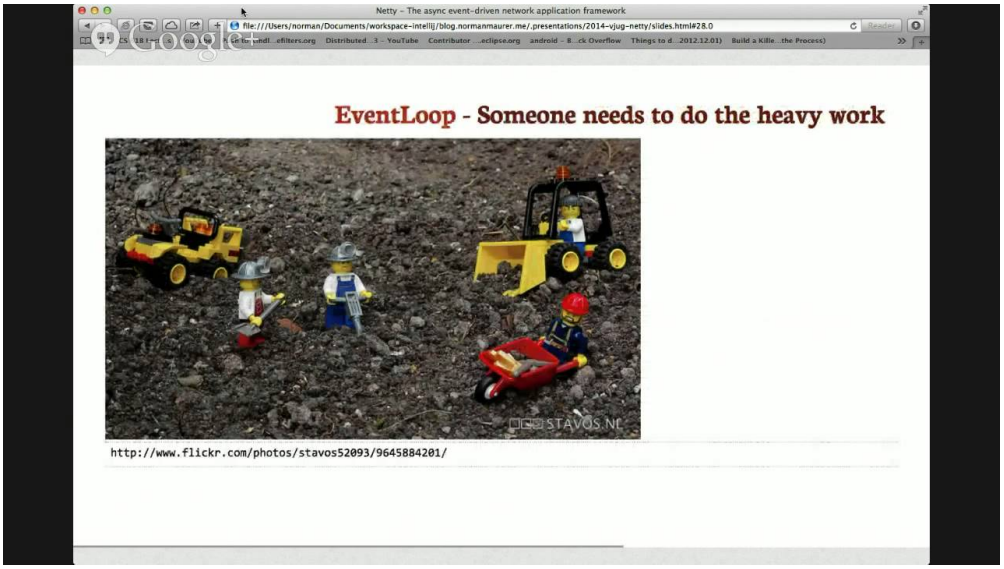
# vJUG Panel: What do the Oracle/Google shenanigans mean to the Java Developer?




Speaker(s): Bruno Souza, Martijn Verburg and Hildeberto Mendonça, Lukas Eder & Michael Rice. (Moderated by Simon Maple)

http://virtualjug.com/?p=865

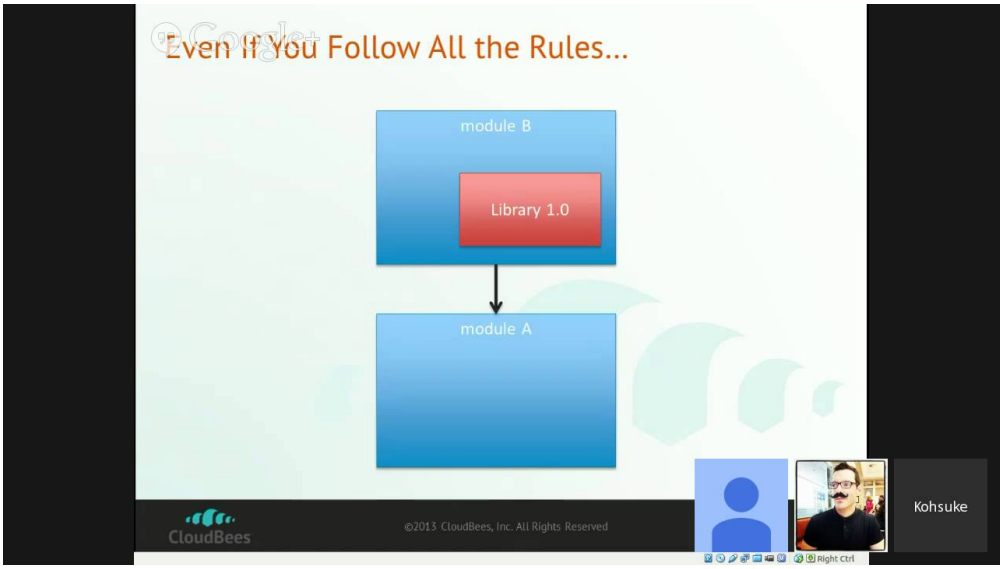
# Netty – The async event-driven network application framework




Speaker: Norman Maurer.

http://virtualjug.com/?p=862

# Evolving code without breaking compatibility

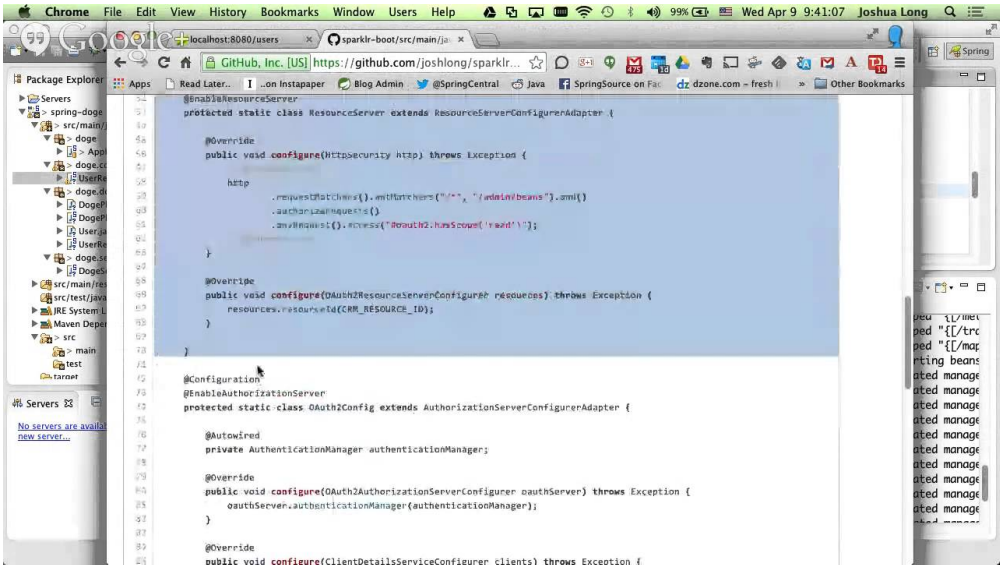



Speaker: Kohsuke Kawaguchi

VJUG

http://virtualjug.com/?p=159

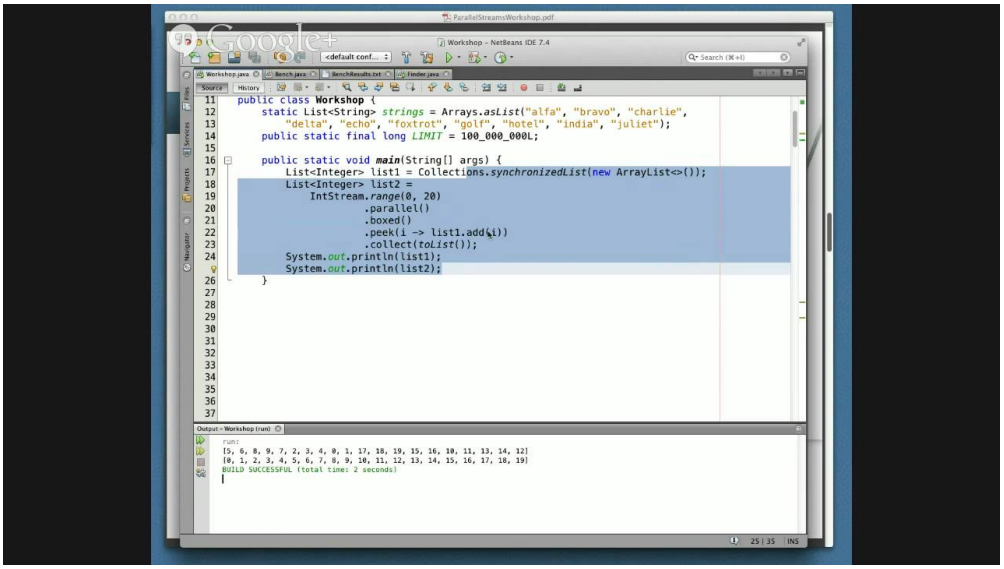
# Building Bootiful Applications with Spring Boot



Speaker: Josh Long

http://virtualjug.com/?p=157

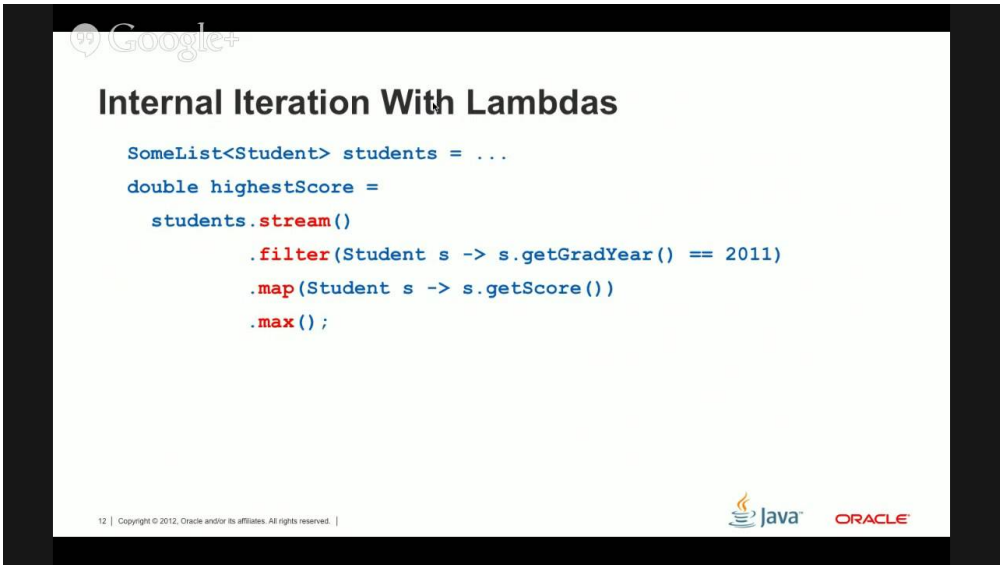
# Java 8 Parallel Streams Workshop




Speaker: Stuart Marks.

http://virtualjug.com/?p=155

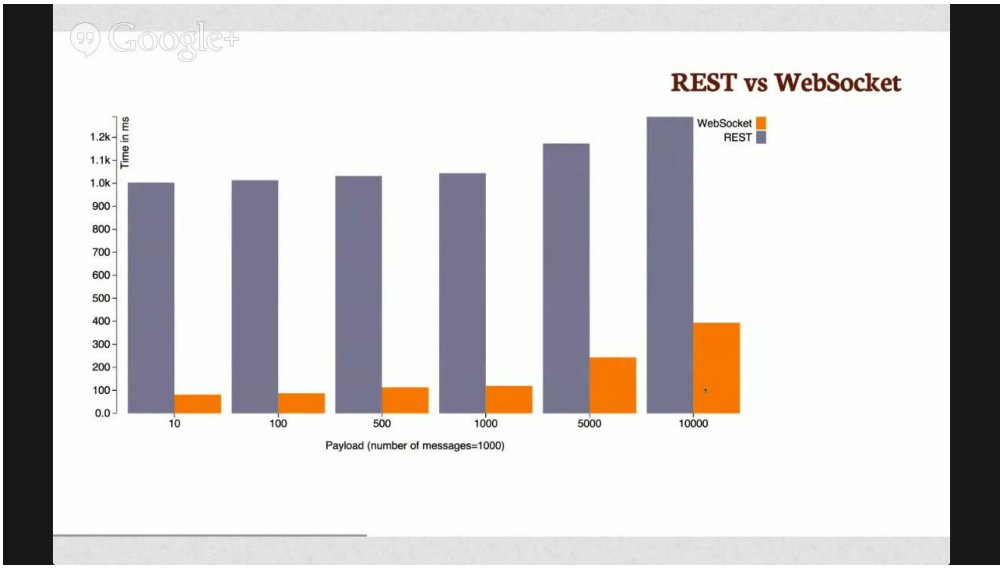
# Project Lambda: Functional Prog. Constructs and Simpler Concurrency in Java SE 8




Speaker: Simon Ritter.

http://virtualjug.com/?p=153

# WebSocket Applications using Java EE 7




Speaker: Arun Gupta.


http://virtualjug.com/?p=151

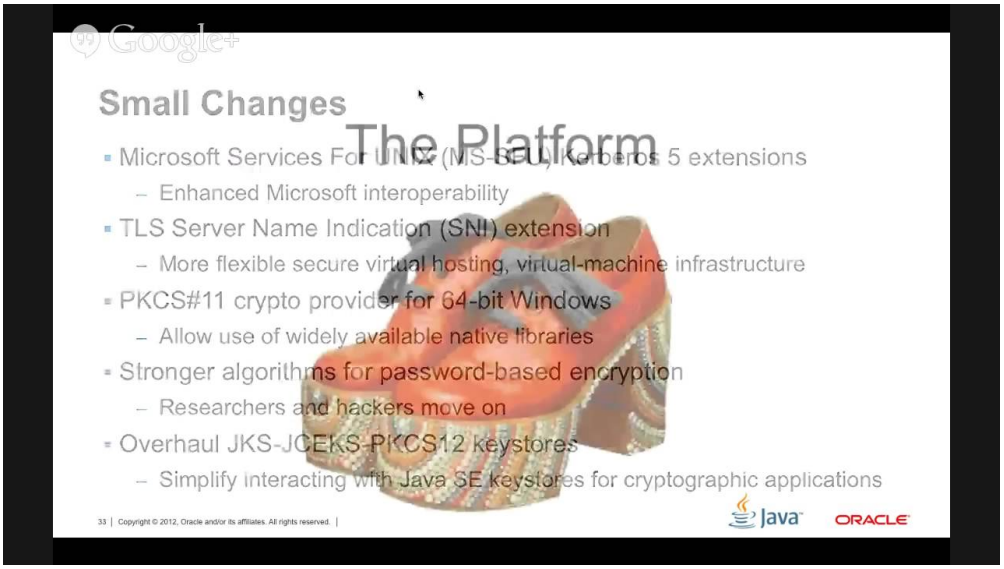
# Comparing JVM Web Frameworks


Speaker: Matt Raible

VJUG

http://virtualjug.com/?p=149

# 55 New Features in Java SE 8




Speaker: Simon Ritter

VJUG

http://virtualjug.com/?p=147

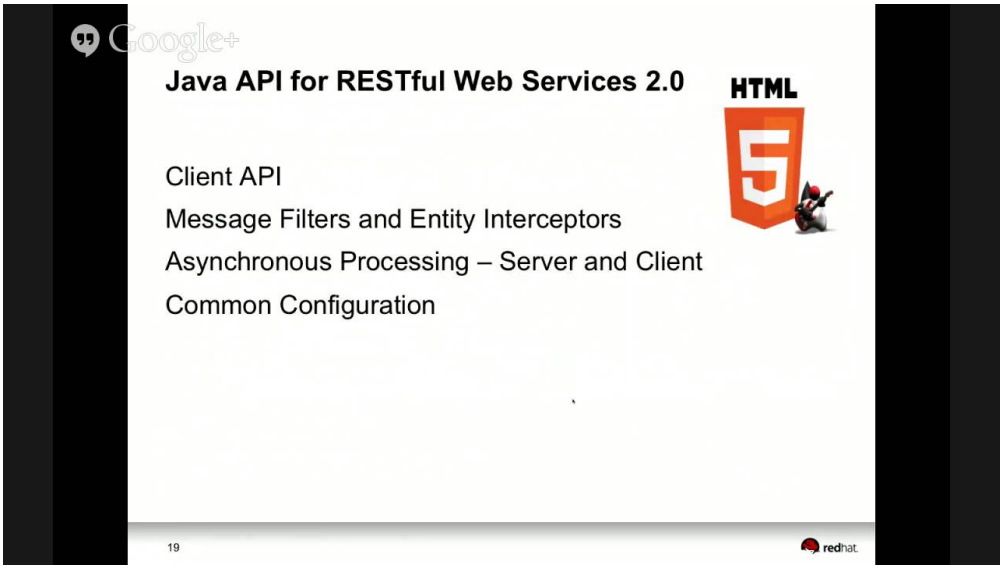
# How To Do Kick-Ass Software Development


Speaker: Sven Peters

VJUG

http://virtualjug.com/?p=144

# Getting started with Java EE 7




Presenter: Arun Gupta.

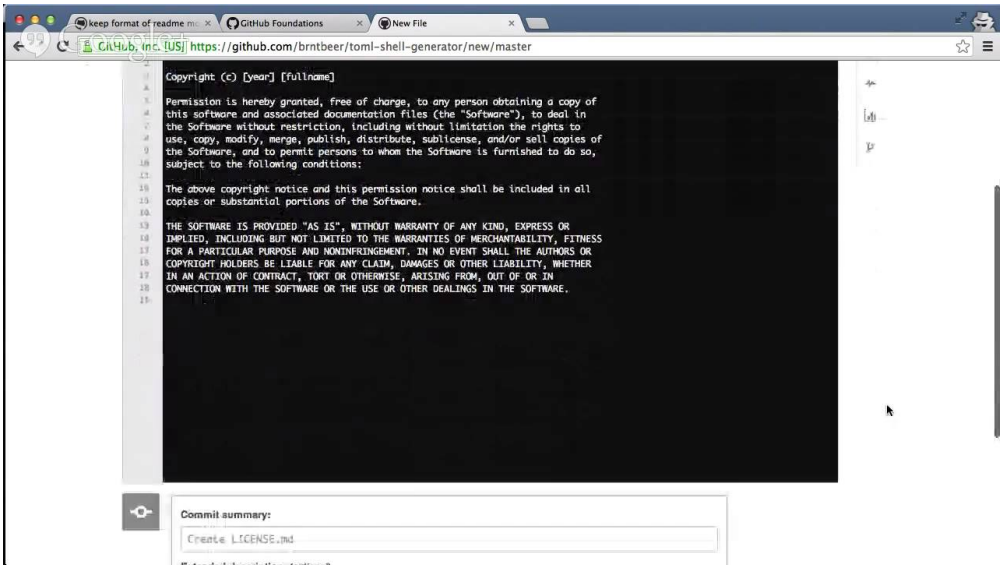

http://virtualjug.com/?p=142

# Don’t be that guy! Developer Security Awareness


Presenter(s): Markus Eisele.


http://virtualjug.com/?p=139

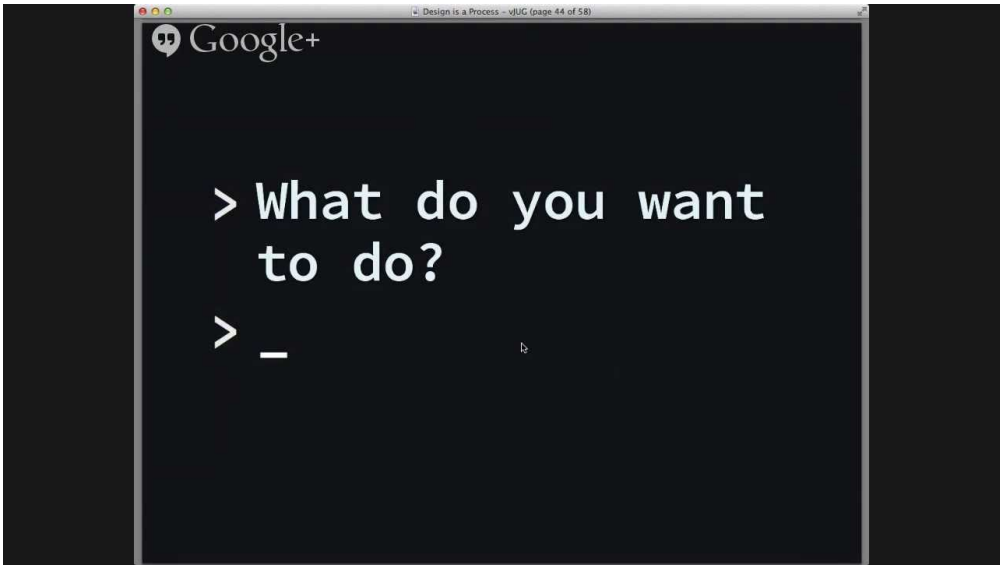
# Drive-by Contributions




Presenter(s): Brent Beer & Matthew McCullough.


http://virtualjug.com/?p=137

# Design is a Process, not a Document




Presenter(s): Trisha Gee.

http://virtualjug.com/?p=132