

Desert Southwest Software Symposium

Sheraton Phoenix Airport Hotel

July 25 - 27, 2008

<http://www.nofluffjuststuff.com/conference/phoenix/2008/07/index.html>

(event schedule as of July 25, 2008)

Fri, Jul. 25, 2008					
	Canyon	Valley	Sky Harbor	University	Rio Salado
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	Know your Java? Venkat Subramaniam	A Thorough Introduction To Groovy Jeff Brown	Test Driven Design Neal Ford	Give it a REST Brian Sletten	The Busy Java Developer's Guide to Concurrency (Part 1: Threads) Ted Neward
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Caring about your Code Quality Venkat Subramaniam	Agile Test Driven Development With Groovy Jeff Brown	Evolutionary SOA Neal Ford	RESTlet for the Weary Brian Sletten	The Busy Java Developer's Guide to Concurrency (Part 2: Concurrency) Ted Neward
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	Towards an Evolutionary Design Venkat Subramaniam	Powerful Metaprogramming Techniques With Groovy Jeff Brown	The Productive Programmer: Practice (10 Ways to Improve Your Code) Neal Ford	Resource-Oriented Computing w/ NetKernel : Software for the 21st Century Brian Sletten	The Busy Java Developer's Guide to ClassLoaders Ted Neward
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	Keynote: by Neal Ford				

Sat, Jul. 26, 2008						
	Sky Harbor	University	Rio Salado	Boardroom	Canyon/Valley	Arizona
8:00 - 9:00 AM	BREAKFAST					
9:00 - 10:30 AM	Regular Expressions in Java Neal Ford	Grails - Agile Web 2.0 The Easy Way Jeff Brown	Tools to facilitate Agile Development Venkat Subramaniam	Viva La Javolution! Brian Sletten	Spring+JPA+Hibernate: Standards Meeting Productivity for Java Persistence Ken Sipe	
10:30 - 11:00 AM	BREAK					
11:00 - 12:30 PM	The Busy Java Developer's Guide to Reflection Ted Neward	Advanced Web Development With Grails Jeff Brown	Acceptance Testing Application Behavior Venkat Subramaniam	What's Going On? : Complex Event Processing w/ Esper Brian Sletten	Spring 2.5 - Spring without XML Ken Sipe	
12:30 - 1:30 PM	LUNCH					
1:30 - 3:00 PM	JavaServer Faces: A Whirlwind Tour David Geary	Code Metrics & Analysis for Agile Projects Neal Ford	The Busy Java Developer's Guide to Debugging Ted Neward	DSL in Groovy Venkat Subramaniam		Architecture and Scaling Ken Sipe
3:00 - 3:15 PM	BREAK					
3:15 - 4:45 PM	Facelets David Geary	"Design Patterns" in Dynamic Languages Neal Ford	The Busy Java Developer's Guide to Monitoring Ted Neward	Got Guice? Venkat Subramaniam		Hacking - The Dark Arts Ken Sipe
4:45 - 5:30 PM	BIRDS OF A FEATHER SESSION					

Sun, Jul. 27, 2008					
	Sky Harbor	University	Rio Salado	Boardroom	Canyon/Valley
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	The Busy Java Developer's Guide to Hacking (on) the JDK Ted Neward	Beginning Object-Relational Mapping with Hibernate Brian Sam-Bodden	Rich Faces David Geary	Iteration 0 Ken Sipe	Productive Programmer: Acceleration & Automation Neal Ford
10:30 - 11:00 AM	MORNING BREAK				
11:00 - 12:30 PM	7 Habits of Highly Effective Developers Ken Sipe	10 ways to use Hibernate effectively Brian Sam-Bodden	Seam David Geary	Better Web Service Modeling and Specification Brian Maso	Productive Programmer: Canonicity & Focus Neal Ford
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	Professional Java UI development with the Eclipse RPC Brian Sam-Bodden	JAX-RS-enabled Brian Maso	Filthy Rich Clients with the Google Web Toolkit, Part I David Geary	The Busy Java Developer's Guide to Java Platform Security Ted Neward	Introduction to JRuby Neal Ford
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	Boosting Programmer productivity with Mylyn Brian Sam-Bodden	Mule-enabled Brian Maso	Filthy Rich Clients with the Google Web Toolkit, Part II David Geary	The Busy Developer's Guide to Scala Ted Neward	Meta-programming JRuby for Fun & Profit Neal Ford

Desert Southwest Software Symposium

Sheraton Phoenix Airport Hotel

July 25 - 27, 2008

<http://www.nofluffjuststuff.com/conference/phoenix/2008/07/index.html>
(event schedule as of July 25, 2008)

Better Web Service Modeling and Specification by Brian Maso

This one's for all the "architects" out there designing and specing services, and those who have to work with them. Whether you are building it or consuming it, the most painful thing about sharing a service is sharing an understanding of what the service does. This presentation teaches you how to dispel ambiguities, techno-mumbo-jumbo, and reliance on institutional knowledge that bogs down service development and testing today using the 5 essential parts of an interface specification known as Operation-State Modeling (OSM).

JAX-RS-enabled by Brian Maso

Introduction to The Java API for RESTful Services (JAX-RS). RESTful Java web services are a pretty radical departure from what you are probably familiar with. JAX-RS avoids the "Java method == service operation" typical in all the popular web service stacks, opting instead for a much more comfortable way of making information services available over HTTP. For the busy developer who wants a fast, practical introduction to RESTful services and the JAX-RS API in particular.

Mule-enabled by Brian Maso

Integrate enterprise resources with the best-known open-source Java ESB. This is an introductory session with a brief summary of Mule internals: the goal is for the Mule-curious to walk away with enough knowledge and techniques to develop Mule-based solutions. You'll have the right start to becoming a Mule development master.

Beginning Object-Relational Mapping with Hibernate by Brian Sam-Bodden

Hibernate is an open source Object-Relational Mapping Framework that mostly automates the tedious and time-consuming task of persisting Java objects to a relational database. Hibernate is quickly becoming the preferred way for enterprise developers to overcome the object-relational impedance mismatch and a good alternative to the coarse-grained Entity EJBs, low-level raw JDBC, and by-committee specifications like JDO. Learn what your choices in the ORM arena, what to look for in an ORM tool, and how to get started with Hibernate for your next J2SE or J2EE project.

10 ways to use Hibernate effectively by Brian Sam-Bodden

Learn 10 tried and true ways to improve the way you use Hibernate today. In this session you would learn about a collection of 10 tips, tricks, practices and tools that will make you more effective at designing, implementing, testing and tuning your application's Hibernate-powered object-relational layer.

Professional Java UI development with the Eclipse RPC by Brian Sam-Bodden

Learn how to build featured rich applications using the Eclipse Rich Client Platform. The Eclipse platform is an open tools platform, on top of this platform you can build your own applications (which do not need to be IDE like or IDE related). Yet you can enjoy the benefits of working with a mature and featured rich platform that can greatly reduce the amount of time required to create a professional-looking and robust Java UI application.

Boosting Programmer productivity with Mylyn by Brian Sam-Bodden

Mylyn is a task-focused toolkit for the Eclipse IDE that allows developers to focus on tasks in a way that they never been able to do before. Mylyn eliminates the constant context switching produced by typical ways IDEs are used. No more scrolling/browsing/searching/tagging/sending emails with progress updates... Mylyn provides a new way of working that allows you to focus on specific tasks by reducing information overload. Mylyn also provides a framework for integrating with the most commonly usage task tracking systems and version control systems. In this talk you'll learn how Mylyn can boost your productivity as a Java developer by letting you get the most out of your IDE.

Give it a REST by Brian Sletten

As developers, we sometimes get to make choices about the technologies we use, sometimes not. We base these decisions on personal experiences, recommendations from others and a general sense of where the

industry is going. Web Services have been all the rage for several years now. We have been told time and again that we should be building systems around them; as an industry, we've never been more confused. Perhaps it is time to Give it a REST.

RESTlet for the Weary by Brian Sletten

If you have started to take a look at REST as way of exposing web services or managing information spaces, you may be frustrated by the support offered by legacy containers. There is no direct support for REST concepts in the J2EE specs (yet). XML-based configurations are so 1990's. Come learn about Restlets, a little API that has caught the attention of many in the RESTafarian community.

Resource-Oriented Computing w/ NetKernel : Software for the 21st Century by Brian Sletten

Imagine the simplicity of REST married to the power of Unix pipes with the benefits of a loosely-coupled, logically-layered architecture. If that is hard to imagine, it may be because the architectures available to you today are convoluted accretions of mismatched technologies, languages, abstractions and data models. NetKernel is a disruptive technology that changes the game. It has been quietly gaining mind share in the past several years; people who are exposed to it don't want to go back to the tired and blue conventions of J2EE and .NET. Not only does it make building the kinds of systems you are building today easier, it does it more efficiently, with less code and a far more scalable runway to allow you to take advantage of the emerging multi-core, multi-CPU hardware that is coming our way. Come see how this open source / commercial product can change the way you think about building software.

Viva La Javolution! by Brian Sletten

You're a good Java programmer. You understand the JDK libraries and how to use them. The problem is that many fundamental APIs don't take the bigger performance picture in mind. Garbage collection can end up killing your app if you aren't careful. Concurrency problems and contention can keep your well-intentioned software from leveraging modern hardware architecture that support multi-core and multi-cpu systems. Who knew that simply using the standard library code the way it was designed was opening you up for performance problems in your apps? Don't worry, Javolution has your back.

What's Going On? : Complex Event Processing w/ Esper by Brian Sletten

We write very complicated software, don't we? In our systems, we detect when simple things happen. Customers log in, people buy things, a stock is sold at a particular price, inventory shifts locations... all of these events mean little things, but what about the larger picture? Complex events are particular patterns of simpler events that suggest something deeper is happening. Do you know how you'd discover these bigger picture occurrences? Come hear how the Esper open source software represents a new class of complex event processing (CEP) frameworks that can be added to even high volume, high transaction systems.

JavaServer Faces: A Whirlwind Tour by David Geary

In April 2005, annual growth rates for jobs in JavaServer Faces, Struts, and Ruby on Rails were all at about 0%. Today, Struts' growth rate still hovers around 0%, but JSF and Rails have taken off. At the end of 2007, both JSF and Rails were growing at a rate of between 400-500% annually (according to indeed.com). JSF has passed the adoption tipping point, and is now the Java-based framework of choice, as is evidenced by its ecosystem. From vendors such as MyEclipse and RedHat to open source projects such as Seam, Facelets, and Ajax4JSF, JSF is where the action is. Come see why JSF is so popular. In this code- and demo-intensive session, I'll show you the fundamentals of JSF. **Prerequisite:** *Some knowledge of Java-based web applications, such as Struts, is a plus, but is not required. If you have a significant experience with JSF, you probably already know most of what's covered in this session.*

Facelets by David Geary

Facelets is a combination of Tiles and Tapestry, and it's the hottest JSF-related open source project on the planet. It's popularity is well deserved, and in fact, much of what is in Facelets today will make its way into the JSF 2.0 spec due out in 2008. So not only can you come to this session and see some really cool demos that you can put to use in the real world, but you'll also be learning JSF 2.0 before it's even been defined! How's that for a ROI? **Prerequisite:** *Some knowledge of JSF is essential. If you're familiar with a templating framework, such as Velocity or Tiles, that's a plus, but not required.*

Rich Faces by David Geary

This talk explores the RichFaces Ajax framework, which is really two frameworks: Ajax4jsf and RichFaces components. In this session you will see how to implement low-level Ajax functionality using Ajax4JSF,

and how to use high-level Ajax components from RichFaces. **Prerequisite:** *Some knowledge of JSF is required, in addition to familiarity with Ajax.*

Seam by David Geary

Have you ever stopped to think that you need to learn two frameworks to develop a non-trivial, database-backed, web application? Struts and iBatis; JSF and Hibernate; Tapestry and EJB3.0. Two frameworks. And then you have to learn to use them together. Why do we have to learn two frameworks just to retrieve "Hello World" from a database and show it in a view. Isn't that crazy? Now you can use one framework, and use one component model. One. Isn't that nice? Seam, a framework built on JSF and EJB3.0, unifies the JSF and EJB component models. Seam is a steam roller, quickly gathering market share among JSF newbies and longtime believers alike. Come see what it's all about. **Prerequisite:** *Some knowledge of JSF is required. If you don't know what a managed bean is, for instance, then attend JSF Whirlwind before this session.*

Filthy Rich Clients with the Google Web Toolkit, Part I by David Geary

The Google Web Toolkit (GWT) is truly a revolutionary framework that lets you develop Ajaxified web applications without knowing anything about Ajax or JavaScript. But the GWT goes way beyond basic Ajax by letting you implement desktop-like applications that run in the ubiquitous browser.

Filthy Rich Clients with the Google Web Toolkit, Part II by David Geary

In the second part of this talk, you will learn how to extend the GWT by implementing custom widgets, including a scrolling viewport and a drag and drop framework. After discussing custom widgets, you will see how to integrate database access into your GWT applications, and how to deploy your GWT applications to external servers.

A Thorough Introduction To Groovy by Jeff Brown

Groovy is an agile dynamic language for the Java platform. The language and its libraries bring many things to the table to ease the process of building applications for the Java platform. This session provides a detailed run through Groovy with lots of code samples to drive home the power of the language.

Agile Test Driven Development With Groovy by Jeff Brown

Dynamic languages bring a lot of interesting elements to the table for teams interested in doing Test Driven Development (TDD). Groovy lends itself very well to TDD and this session demonstrates many features of the language and its libraries that help teams build more testable systems and build better tests.

Powerful Metaprogramming Techniques With Groovy by Jeff Brown

Metaprogramming is a key component in building truly dynamic and flexible applications with Groovy. Groovy's metaprogramming capabilities bring great new possibilities to the table that would be very difficult or just plain impossible to write with Java alone. This session will demystify a lot of the magic that seems to be going on inside of a Groovy application. **Prerequisite:** *A Thorough Introduction To Groovy*

Grails - Agile Web 2.0 The Easy Way by Jeff Brown

Grails is a full stack MVC framework for building web applications for the Java platform. Grails makes web application development both fun and easy. This session covers all of the fundamentals of building web applications with Grails.

Advanced Web Development With Grails by Jeff Brown

Grails makes web application development both fun and easy. This session dives beyond the basics to cover advanced details of Grails that bring the really exciting features to your applications. **Prerequisite:** *Grails - Agile Web 2.0 The Easy Way*

Spring+JPA+Hibernate: Standards Meeting Productivity for Java Persistence by Ken Sipe

Well the standards created EntityBeans.... yea. and the community created Hibernate. Fortunately the standards body learned some lessons and created JPA. JPA requires a vendor implementation and none make a better choice than Hibernate. Combined with Spring this trio is a powerhouse when it comes to developer productivity on applications requiring persistence.

Spring 2.5 - Spring without XML by Ken Sipe

Spring 2.5 is brand spanking new, with a number of fantastic features. With growth of large and complex Spring applications which struggle with xml manageability and with the added pressure of Guice and SEAM there is a push for less XML, with solution leaning towards annotations. Spring 2.5 adds to the toolset provided in Spring 2.0 to provide a development environment where XML is greatly reduced... or eliminated if you so choose.

Architecture and Scaling by Ken Sipe

Scale... what is scale... how do you applications which are scalable. How do you know if the application scales?

Hacking - The Dark Arts by Ken Sipe

A live Hacking demonstration exposing the tools and techniques used by Hackers.

Iteration 0 by Ken Sipe

The success of an Agile / SCRUM project is a successful start. The first interaction is often referred to as iteration 0. Other iterations have a set of stories with clear acceptance criteria which establishes the velocity of the team and its effort. What then is accomplished in iteration 0? How do we get an Agile process started.

7 Habits of Highly Effective Developers by Ken Sipe

Thoughts lead to words, words lead to action, actions lead to habits. In this session we'll sharpen the development saw in the process of understanding what makes a hyper-productive programmer. The focus will consist of developer habits and development processes.

Test Driven Design by Neal Ford

Most developers think that "TDD" stands for Test-driven Development. But it really should stand for "Test-driven Design". Rigorously using TDD makes your code much better in multiple ways.

Evolutionary SOA by Neal Ford

This session demonstrates that "Agility" and "SOA" complement each other quite well. Just because SOA is buzz-word compliant doesn't mean that you should throw good practices out the window. This session demonstrates how you can apply the principles of agility to building highly complex distributed enterprises.

The Productive Programmer: Practice (10 Ways to Improve Your Code) by Neal Ford

No one writes perfect code: even the best developers fall into bad habits and traps. These topics from The Productive Programmer illustrate blind spots and helps you write better code.

Keynote: Ancient Philosophers & Blowhard Jamborees by Neal Ford

It turns out that ancient philosophers knew a lot about software -- did you know that Plato defined object-oriented programming? This keynote applies old lessons to new problems and old problems to new lessons. It describes why SOA is so hard, and why people in your company make bone-headed decisions. What other keynote includes Rube Goldberg, Aristotle, Dave Thomas, and Chindia?

Regular Expressions in Java by Neal Ford

Regular expressions should be an integral part of every developer's toolbox, but most don't realize what an important topic it is. Regular expressions have existed for decades, but many developers don't understand how to take full advantage of this powerful mechanism, either through command line tools and editors or in their development.

Code Metrics & Analysis for Agile Projects by Neal Ford

What does code + methodology have to do with one another? Everything! Agile projects focus on delivering working code, and tools exist to allow you to verify some quality metrics for your code. This session is a survey of tools and metrics that allow you to determine the quality of your code and strategies to "wire it" into your agile project.

"Design Patterns" in Dynamic Languages by Neal Ford

The Gang of Four book should have been entitled "Palliatives for Statically Typed Languages", because the recipes it provides are cumbersome solutions to the problems it poses. Using powerful languages makes the

solutions in the GoF book look hopelessly complicated. This session shows how to solve the same problems concisely, elegantly, and with far fewer lines of code using the facilities of dynamic languages.

Productive Programmer: Acceleration & Automation by Neal Ford

Developers from the 1980s would be shocked at how inefficiently developers use their computers because of the advent of graphical operating systems. This talk describes how to reclaim productivity afforded by intelligent use of command lines and other ways of accelerating your interaction with the computer and bending computers to do your bidding. Stop working so hard for your computer!

Productive Programmer: Canonicity & Focus by Neal Ford

Getting work done in modern office environments is a daunting task. This session tackles 2 of the things that drag down developer productivity: lack of focus and creeping repetition.

Introduction to JRuby by Neal Ford

This session describes JRuby, the 100% pure-Java implementation of the Ruby programming language. It covers the basics of programming with JRuby and examples of how to integrate it into existing Java projects.

Meta-programming JRuby for Fun & Profit by Neal Ford

Ruby is the revenge of the Smalltalkers. Not since Smalltalk has a language had such powerful meta-programming facilities. While this may seem like a minor feature, it turns out that surgical meta-programming allows solutions to problems that are clearer, more concise, more maintainable, and take orders of magnitudes fewer lines of code.

The Busy Java Developer's Guide to Concurrency (Part 1: Threads) by Ted Neward

Java's threading capabilities took a serious turn for the better with the release of Java5, thanks to the incorporation of the `java.util.concurrent` packages, a set of pre-built components for thread pooling and execution, synchronization, and more.

The Busy Java Developer's Guide to Concurrency (Part 2: Concurrency) by Ted Neward

Java's threading capabilities have been a part of the Java platform since its inception, yet for many Java developers, using Threads still remain a dark and mysterious art, and synchronization beyond the use of the "synchronized" keyword is almost unknown. *Prerequisite: The Busy Java Developer's Guide to Concurrency (Part 1: Threads)*

The Busy Java Developer's Guide to ClassLoaders by Ted Neward

If you've ever gotten a `ClassCastException` and just knew the runtime was wrong about it, or found yourself copying `.jar` files all over your production server just to get your code to run, then you probably find the Java `ClassLoader` mechanism to be deep, dark, mysterious, and incomprehensible. Take a deep breath, and relax--`ClassLoaders` aren't as bad as they seem at first, once you understand a few basic rules regarding their operation, and have a bit more tools in your belt to diagnose `ClassLoader` problems. And once you've got that, and hear about `ClassLoaders`' ability to run multiple versions of the same code at the same time, and to provide isolation barriers inside your application, or even compile code on the fly from source form, you might just find that you like `ClassLoaders` after all... maybe.

The Busy Java Developer's Guide to Reflection by Ted Neward

If you've never used Reflection (`java.lang.reflect`), you don't know what you're missing. In this presentation, we'll take a code-first, soup-to-nuts look at the Java Reflection APIs, from how to examine the class metadata that Reflection provides, to using annotations to enhance that metadata with your own information, even through the use of Java Dynamic Proxies to create flexible object "interceptors" that can layer services in front of ordinary method calls with nothing more complicated and an interface and a factory.

The Busy Java Developer's Guide to Debugging by Ted Neward

Bugs? We all know your code has no bugs, but someday, you're going to find yourself tracking down a bug in somebody else's code, and that's when it's going to be helpful to have some basic ideas about bug-tracking in your toolbox. Learn to make use of the wealth of tools that the Java Standard Platform

makes available to you--tools that your IDE may not know exist, tools that you can make use of even within a production environment.

The Busy Java Developer's Guide to Monitoring by Ted Neward

Crashes? Outages? Slow response? We all know that it's never your code that causes these things, but for some reason those pesky system administrators still insist on paging you at 4AM to come in and fix those problems, anyway. For some reason, they just keep expecting you to support this thing, even after QA said it was OK!

The Busy Java Developer's Guide to Hacking (on) the JDK by Ted Neward

Ever since its 1.1 release, the Java Virtual Machine steadily becomes a more and more "hackable" (configurable, pluggable, customizable, choose your own adjective here) platform for Java developers, yet few, if any, Java developers take advantage of it. Time to take the kid gloves off, crack open the platform, and see what's there. Time to play.

The Busy Java Developer's Guide to Java Platform Security by Ted Neward

Permissions, policy, SecurityExceptions, oh my! The Java platform is a rich and powerful platform, complete with a rich and powerful security mechanism, but sometimes understanding it and how it works can be daunting and intimidating, and leave developers with the basic impression that it's mysterious and dark and incomprehensible. Nothing could be further from the truth, and in this presentation, we'll take a pragmatic, code-first look at the Java security platform, including Permissions, the SecurityManager and its successor, AccessController, the Policy class and policy file syntax, JAAS, and more.

The Busy Developer's Guide to Scala by Ted Neward

Scala is a new programming language incorporating the most important concepts of object-oriented and functional languages and running on top of the Java Virtual Machine as standard "dot-class" files.

Know your Java? by Venkat Subramaniam

Java has been around for well over a decade now. It started out with the goal of being simple. Over the years, its picked up quite a bit of features and along comes complexity. In this presentation we will take a look at some tricky features of Java, those that can trip you over, and also look at some ways to improve your Java code.

Caring about your Code Quality by Venkat Subramaniam

We all have seen our share of bad code. We certainly have come across some good code as well. What are the characteristics of good code? How can we identify those? What practices can promote us to write and maintain more of those good quality code. This presentation will focus on this topic that has a major impact on our ability to be agile and succeed.

Towards an Evolutionary Design by Venkat Subramaniam

A good design is critical for success with agile development. That does not mean a big up-front design. The design has to be evolutionary. However, the design you evolve must be extensible and maintainable. After all, you can't be agile if your design sucks.

Tools to facilitate Agile Development by Venkat Subramaniam

The first item in the Agile Manifesto reads that we must prefer "people and interaction over process and tools." Given a choice between average people with superior tools and superior people with average tools, you are likely to achieve greater success with the latter. However, it is important to be continuous and not be episodic?so you want to get continuous feedback about the state, health, and quality of your code and application. Tools can help us a great deal to realize this and make us productive.

Acceptance Testing Application Behavior by Venkat Subramaniam

How do you ensure your applications meet the expectations of your key customers? In this session we will explore using the FIT tool and Behavior Driven Design tools to do exactly this.

DSL in Groovy by Venkat Subramaniam

DSL or Domain Specific Languages focus on a domain or problem at hand. They're expressive, but their restricted scope keeps them simple and small from the user point of view. However, designing them is not

easy. In this presentation we will explore the features of Groovy and show how they can be used to create DSLs.

Got Guice? by Venkat Subramaniam

In this presentation we will take a look at Google's dependency injection framework, discuss its features, capabilities, strengths, and weakness. We will then discuss where it stands in comparison to Spring.