

New England Software Symposium

Crowne Plaza North Shore

March 5 - 7, 2010

The No Fluff Just Stuff Java Symposium Series is proud to announce the return of the New England Software Symposium on March 5 - 7, 2010. NESS will be held at the Crowne Plaza North Shore.

Since 2001, the No Fluff Just Stuff Java Symposium has been regarded as the premier Java/Agility event series anywhere serving over 29,500 attendees with some 190 events. The popularity of the NFJS symposium series can be traced to the following:

- 1). Exceptional Speakers
- 2). Limited Attendance - capped at 250 people
- 3). No Vendors, No Sales Pitches, No Marketecture
- 4). Excellent networking opportunity with speakers and fellow attendees because of small size.
- 5). The Best Value in the Java conferencing space period.

Topics for the 2010 NFJS Tour

Languages on the JVM: Groovy, JRuby, Clojure, Scala
HTML5

Enterprise Java

Core Java, JVM Internals

No SQL: MongoDB

JSF, GWT

Agility

Groovy, Grails, Gradle

REST, RDFa, Resource Oriented Architectures

GIT Version Control

jQuery, Ajax, Flex, RIA

Mobile Applications - iPhone and Android

More...

The Registration Fee Includes:

3 Day All Access Pass to NESS

All Meals/Snacks - duration of the symposium

90 Day IntelliJ license compliments of JetBrains

Session Materials

Custom NFJS Binder

Great Giveaways @ NFJS

Early Bird Registration: \$850/person good thru 2/19/10 after \$950

Excellent Group Discounts Available - bring your entire development team to the show - no travel required!! Rate good thru 2/19/10

Registration Fees

Attendees	Before Feb. 19, 2010	After Feb. 19, 2010
5-9	\$750	\$850
10-14	\$725	\$825
15-24	\$700	\$800
25+	\$675	\$775

Go to <http://www.nofluffjuststuff.com/conference/boston/2010/03/home> and register today!

New England Software Symposium

Crowne Plaza North Shore

March 5 - 7, 2010

Fri, Mar. 05, 2010					
	North Shore A	Marblehead	Newburyport	Gloucester	Ipswich
12:00 - 1:00 PM	REGISTRATION				
1:00 - 1:15 PM	WELCOME				
1:15 - 2:45 PM	Implementing Evolutionary Architecture Neal Ford	Spring 3 into REST Ken Sipe	Intro to Messaging Using JMS and ActiveMQ Mark Richards	Encryption on the JVM: Boot Camp Matthew McCullough	Slaying the Legacy Dragon: Practical Lessons in Replacing Old Software Tim Berglund
2:45 - 3:15 PM	BREAK				
3:15 - 4:45 PM	Implementing Emergent Design Neal Ford	Enter The Gradle Ken Sipe	The Art of Messaging Mark Richards	Encryption on the JVM: Advanced Techniques Matthew McCullough	Decision Making in Software Teams Tim Berglund
4:45 - 5:00 PM	BREAK				
5:00 - 6:30 PM	Testing the Entire Stack Neal Ford	Debugging your Production JVM Ken Sipe	Using Apache Camel Mark Richards	Hadoop: Divide and Conquer Gigantic Datasets (Intro) Matthew McCullough	Open Source Business Intelligence Tim Berglund
6:30 - 7:15 PM	DINNER				
7:15 - 8:00 PM	KEYNOTE: Neal Ford & Martin Fowler - Why Agile Development Works (Not How)				

Sat, Mar. 06, 2010					
	North Shore A	Marblehead	Newburyport	Gloucester	Ipswich
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	XSS-Proof Ken Sipe	Architect for Scale Michael Nygard	iBeans: The Simplest Service Integrations You've Ever Implemented Matthew McCullough	The Busy Java Developer's Guide to Collections Ted Neward	Practical Agile Database Development Tim Berglund
10:30 - 11:00 AM	BREAK				
11:00 - 12:30 PM	Agile Engineering Practices Neal Ford	Software Architecture for the Cloud Michael Nygard	Open Source Debugging Tools for Java Matthew McCullough	The Busy Java Developer's Guide to Functional Java Ted Neward	Concurrency Revolution: The Hardware Story Brian Goetz
12:30 - 1:30 PM	LUNCH				
1:30 - 3:00 PM	What's Brewing in Java Venkat Subramaniam	High Performance Persistence with Redis Michael Nygard	Migrating to Maven 3.0 Matthew McCullough	The Busy Java Developer's Guide to Advanced Collections Ted Neward	Towards a Universal VM Brian Goetz
3:00 - 3:15 PM	BREAK				
3:15 - 4:45 PM	HTML 5 ... and the Kitchen Sink Brian Sletten	Design for Operations Michael Nygard	How to Approach Refactoring Venkat Subramaniam	MongoDB Ted Neward	Stupid JIT Tricks Brian Goetz
4:45 - 5:45 PM	300 Series & Refactoring Workshop with Venkat Subramaniam				

Sun, Mar. 07, 2010					
	North Shore A	Marblehead	Newburyport	Gloucester	Ipswich
8:00 - 9:00 AM	BREAKFAST				
9:00 - 10:30 AM	jQuery: Ajax Made Easy Nathaniel Schutta	Performance and Scalability Revisited: In-Memory Data Grids Aleksandar Seovic	The Java Memory Model Brian Goetz	REST : Information-Driven Architectures for the 21st Century Brian Sletten	Programming Scala Venkat Subramaniam
10:30 - 11:00 AM	MORNING BREAK				
11:00 - 12:30 PM	JavaScript Beyond the Basics Nathaniel Schutta	In-Memory Data Grids: Not Your Mom's Cache Aleksandar Seovic	Are All Web Applications Broken? Brian Goetz	RDFA : Weaving Richness and Meaning in the Web Brian Sletten	Scala Tricks Venkat Subramaniam
12:30 - 1:15 PM	LUNCH				
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION				
2:15 - 3:45 PM	Agile UI Nathaniel Schutta	Testing with dependencies Venkat Subramaniam	Cloud computing deep dive for Google App Engine and Amazon EC2 Rohit Bhardwaj	SPARQL : Querying the Web of Data Brian Sletten	JRuby in Practice Aaron Bedra
3:45 - 4:00 PM	BREAK				
4:00 - 5:30 PM	Hacking Your Brain for Fun and Profit Nathaniel Schutta	Transforming to Groovy Venkat Subramaniam	Android mobile application development: Cool apps that surprise and delight mobile users—built by developers like you Rohit Bhardwaj	Semantic SOA : Meaningful Service Strategies Brian Sletten	The Art of the Spike Aaron Bedra

New England Software Symposium

Crowne Plaza North Shore

March 5 - 7, 2010

JRuby in Practice by Aaron Bedra

Ruby has made a significant upward trend in the past few years. Alongside this trend Charles Nutter and the fantastic JRuby team have implemented a version of Ruby that runs on the JVM giving you the power of Ruby coupled with the advantages of running on the JVM. Come see for yourself how you can harness the power of rapid development in Ruby and still maintain all the Java interoperability you need to help you build on top of your existing systems.

The Art of the Spike by Aaron Bedra

Exploring new technologies can be both challenging and rewarding. A good spike can make or break a new feature for your application. Have you ever thought that a technology or practice your company isn't currently using is the perfect fit for your next iteration? This is your time to shine! In this session you will learn how to treat new technologies as first class citizens and prove that they fit your needs. You will also learn how to provide concrete evidence supporting your decision. By the end of this session your fear of introducing new technology will simply melt away.

Slaying the Legacy Dragon: Practical Lessons in Replacing Old Software by Tim Berglund

It's a given that everyone hates the legacy application and wants to replace it. You're tired of the brittle, untested code, the outdated frameworks, the platform nobody cares about anymore. You want to apply current practices and the productivity gains of today's tools. Usually this is just a frustrated dream, but every once in a while, you actually get to do it. That's great news, but it raises a question: how do you...do that?

Decision Making in Software Teams by Tim Berglund

Alistair Cockburn has described software development as a game in which we choose among three moves: invent, decide, and communicate. Most of our time at No Fluff is spent learning how to be better at inventing. Beyond that, we understand the importance of good communication, and take steps to improve in that capacity. Rarely, however, do we acknowledge the role of decision making in the life of software teams, what can cause it to go wrong, and how to improve it.

Open Source Business Intelligence by Tim Berglund

Traditionally, business intelligence tools have been a high-cost part of any enterprise's software inventory. Recently, options have emerged that allow architects to build a credible business intelligence stack out of entirely open-source components. In this brief overview, we will demonstrate ETL, reporting, and analytics tool that can be deployed free or at low cost. Learn how to turn your company's transactional database into a rich data asset with a business-friendly user interface that integrates into your existing software infrastructure.

Practical Agile Database Development by Tim Berglund

Do your team's agile practices extend to the database? Agile methods are fairly well-understood as they apply to code, but these principles are not commonly understood or practiced on the databases that typically accompany enterprise software projects. Learn the tools, techniques, and mindset your team needs to make incremental improvements to the database's design over time with confidence.

Cloud computing deep dive for Google App Engine and Amazon EC2 by Rohit Bhardwaj

In this session we will take a deep dive at few cloud computing examples from real world and participants will be able to know how to use cloud computing for Google App Engine, Amazon EC2 and few others.

Android mobile application development: Cool apps that surprise and delight mobile users—built by developers like you by Rohit Bhardwaj

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. Cool apps that surprise and delight mobile users—built by developers like you—are a huge part of the Android vision. In this presentation we will explore many examples of android. **Prerequisite:** none

Implementing Evolutionary Architecture by Neal Ford

This talk describes an agile approach to architecture, and merges the current state-of-the-art thinking in both service oriented architectures(SOA) and web-based architectures like HTTP and REST.

Implementing Emergent Design by Neal Ford

Emergent design is a big topic in the agile architecture and design community. This session covers some of the theory behind emergent design, but spends most of its time showing examples of how you can implement this important concept. **Prerequisite:** *understanding of architectural and design concepts*

Testing the Entire Stack by Neal Ford

This talk covers all aspects of software testing, starting with unit testing, then move into functional and integration testing, and finishing off with user acceptance testing. I show examples of tools in the Java universe to make it easy and effective at each testing level, and cover what kinds of tests should exist at what part of the ecosystem. **Prerequisite:** *Confusion about what to test when and where*

Agile Engineering Practices by Neal Ford

Most of the time when people talk about agile software development, they talk about project and planning practices and never mention actual development practices. This talk delves into best development practices for agile projects, covering all of its aspects. **Prerequisite:** *Having worked in an organization that values bureaucracy more than individuals*

Concurrency Revolution: The Hardware Story by Brian Goetz

Do software developers need to know anything about CPU architecture? They do if they aspire to be performance experts. Modern CPUs behave almost nothing like the sequential Von Neumann machine model we know and love. This session provides an overview of the architecture of modern CPUs, how this has changed in recent years, and what the implications are for software development and performance management.

Towards a Universal VM by Brian Goetz

The success of the Java platform is powered by the Java Virtual Machine (JVM), which many people assume is tied to the Java programming language. In fact, 100+ programming languages are hosted on the JVM, including JavaFX, JRuby, Jython, Groovy, Clojure, and Scala. A key implementation challenge is to make code written in non-Java languages run as fast as code written in the Java language.

Stupid JIT Tricks by Brian Goetz

Ever wondered what happens to your bytecodes when they're executed by a Java Virtual Machine? This talk provides a peek "under the hood" of modern JVMs, exploring dynamic compilation, speculative optimization, garbage collection, and some hardware-specific optimizations.

The Java Memory Model by Brian Goetz

What's the worst thing that can happen when you fail to synchronize in a concurrent Java program? Its probably worse than you think -- modern shared-memory processors can do some pretty weird things when left to their own devices.

Are All Web Applications Broken? by Brian Goetz

Many developers believe that web frameworks "take care of" the details of concurrency, but this is only because most web applications make limited use of state. Stateful web applications also need to be careful about hazards like races. This talk will use the Java Memory Model to analyze common patterns of state management in web applications. **Prerequisite:** *The Java Memory Model*

Encryption on the JVM: Boot Camp by Matthew McCullough

Does your application transmit customer information? Are there fields of sensitive customer data stored in your DB? Can your application be used on insecure networks? If so, you need a working knowledge of encryption and how to leverage Open Source APIs and libraries to make securing your data as easy as possible. Encryption is quickly becoming a developer's new frontier of responsibility in many data-centric applications.

Encryption on the JVM: Advanced Techniques by Matthew McCullough

Now that you have the basics of encryption under your belt, we'll advance to talking about where it is sensible and performant to add this level of security to your application. Symmetric key and public key encryption have various levels of processing overhead, so you can't blindly just use the "best" encryption out there. What about password hashes? Did you know they are vulnerable with our "salt"? **Prerequisite:** *Encryption Bootcamp on the JVM*

Hadoop: Divide and Conquer Gigantic Datasets (Intro) by Matthew McCullough

Moore's law has finally hit the wall and CPU speeds have actually decreased in the last few years. The industry is reacting with hardware with an ever-growing number of cores and software that can leverage "grids" of distributed, often commodity, computing resources. But how is a traditional Java developer supposed to easily take advantage of this revolution? The answer is the Apache Hadoop family of projects. Hadoop is a suite of Open Source APIs at the forefront of this grid computing revolution and is considered the absolute gold standard for the divide-and-conquer model of distributed problem crunching. The well-travelled Apache Hadoop framework is currently being leveraged in production by prominent names such as Yahoo, IBM, Amazon, Adobe, AOL, Facebook and Hulu just to name a few.

iBeans: The Simplest Service Integrations You've Ever Implemented by Matthew McCullough

No app is an island nowadays and your bleeding edge Java & JavaScript apps demand that you integrate with Facebook, Amazon, Gmail, Google Search, Twitter or S3 just to name a few. Make your next integration project a breeze by leveraging the successful work of others from the iBeans Central repository, or if necessary, simply author a new iBean and contribute it back for the benefit of all. iBeans a new ultra-light service integration framework written in Java, but targeting both Java and JavaScript. It provides a centralized mechanism for community contributions of beans to the most commonly used services such as Twitter, Flickr, Gmail and more.

Open Source Debugging Tools for Java by Matthew McCullough

This session will survey a wide range of tools across the Java space. We'll look at utilities such as VisualVM, jstatd, jps, jhat, jmap, Eclipse Memory Analyzer, jtracert, btrace and more. Open Source is not just a suite of libraries you consume within your application, but now reaches into the space of tools to help you troubleshoot and improve your applications. The price of these tools eliminates barriers to their use and their open source nature allows you to mix and match them into compositions that work well for your application's unique debugging needs.

Migrating to Maven 3.0 by Matthew McCullough

Explore what's new on the cutting edge release of Maven, version 3.0. We'll explore the performance improvements, features that make debugging Maven issues easier, and changes to POMs that may require modifications to your build, but will result in more determinate build outputs.

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

For so many Java developers, the `java.util.*` package consists of `List`, `ArrayList`, and maybe `Map` and `HashMap`. But the Collections classes are so much more powerful than many of us are led to believe, and all it requires is a small amount of digging and some simple exploration to begin to "get" the real power of the Collection classes.

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

Much noise has been made in recent years about functional languages, like Scala or Haskell, and their benefits relative to object-oriented languages, most notably Java. Unfortunately, as wonderful as many of those benefits are, the fact remains that most Java developers will either not want or not be able to adopt those languages for writing day-to-day code. Which leaves us with a basic question: if I can't use these functional languages to write production code, is there any advantage to learning about them? The short answer is yes, for the fundamental premise--"I can't use functional code on my Java project"--is flawed. Java developers can, in fact, make use of functional ideas, and what's better, they don't even have to reinvent them for Java--thanks to the `FunctionalJava` library, many of the core primitives--interfaces that serve as base types for creating function values, for example--already exist, ready to be used.

The Busy Java Developer's Guide to Advanced Collections by Ted Neward

Once you've learned the core Collections classes, you're done, right? You know everything there is to know about Collections, and you can "check that off" your list of Java packages you have to learn and know, right?

Prerequisite: *Busy Java Developer's Guide to Collections*

MongoDB by Ted Neward

MongoDB is designed for problems without heavy transactional requirements that aren't easily solved by traditional RDBMSs, including problems which require the database to span many servers. Like other document-oriented database systems such as CouchDB, MongoDB is not a relational database management system. The database manages collections of JSON-like documents which are stored in a binary format referred to as BSON.

Architect for Scale by Michael Nygard

Is your system small, medium, large, or super-size? Is traffic on it's way up? Architecture patterns and structures that work at one scale seldom work across all of them. A communication style that's appropriate for small websites will probably fail badly if you apply it to world-wide networks of computers. Likewise, structures that work for large-scale systems are probably too complex and expensive to be worth it for small sites.

Software Architecture for the Cloud by Michael Nygard

Servers, storage, networking, backups... they're all vanishing into the "clouds". Cloud Computing is the emerging architecture for massive, scalable infrastructure that your company doesn't have to own or operate. In this session, Michael will discuss the ingredients of real cloud computing and how you can apply it to your applications. He will show several architectures and discuss applications that fit each of these models. Finally, he will also talk about some of the pitfalls and problems that cloud computing customers can encounter.

High Performance Persistence with Redis by Michael Nygard

Redis is one of the fresh crop of "NoSQL" storage solutions. It's a distributed key-value store that knows how to deal with data structures. Oh, and it happens to also be incredibly fast. Like, microseconds per write.

Design for Operations by Michael Nygard

If your software fails in production, nobody will care how great the development project was, or how well the system passed QA. Production operations, the domain of your systems' least-appreciated stakeholders, is where the rubber meets the road. Come learn how to build your systems to thrive in Operations.

Intro to Messaging Using JMS and ActiveMQ by Mark Richards

More and more companies are using messaging as a means for heterogeneous communication, scalability, performance, and load balancing. Why? Because messaging provides asynchronous requests, guaranteed delivery, load balancing, and ease of development. In this session I will introduce some basic messaging fundamentals, then show how easy it is to send and receive messages using the JMS API. During this session I will also show how to setup and configure ActiveMQ, an open source enterprise-wide messaging provider. By attending this session you will see how easy messaging using JMS really is!

The Art of Messaging by Mark Richards

Messaging is both a science and an art. Messaging is a science with respect to the mechanics of the JMS API and the syntax for sending and receiving messages. However, messaging is also an art when it comes to applying the JMS API to solve real-world problems. In this session I will review some of the more common use cases for messaging and show techniques for significantly increasing both the performance and scalability of messaging-based applications. Using ActiveMQ, you will see how to create embedded brokers, solve internal application bottleneck issues, how to use asynchronous logging with Log4J and JMS, and how to significantly speed up your messaging applications. In this session I will also describe and demonstrate some emerging trends in RESTful JMS (that is, JMS over HTTP). Come to this session to find out how much fun messaging can really be! **Prerequisite:** *Some knowledge of JMS and Messaging in general*

Using Apache Camel by Mark Richards

Apache Camel is a robust open source integration framework that handles routing and mediation tasks associated with enterprise integration. Camel allows you to quickly and easily route messages and integrate components in a distributed, decoupled manner. For example, using the Camel Java DSL, you can send

and receive JMS messages in just a couple of lines of Java code. In this live coding session I will describe what Camel is, describe the overall architecture, show why it is useful, and demonstrate through live coding examples how to use the Camel Java DSL to write simple (and complex) routing logic. By attending this session you will learn Camel well enough to use it at work the next day.

jQuery: Ajax Made Easy by Nathaniel Schutta

Sure, Ajax might not be the hardest thing you'll have to do on your current project, but that doesn't mean we can't use a little help here and there. While there are a plethora of excellent choices in the Ajax library space, jQuery is fast becoming one of the most popular. In this talk, we'll see why. In addition to its outstanding support for CSS selectors, dirt simple DOM manipulation, event handling and animations, jQuery also supports a rich ecosystem of plugins that provide an abundance of top notch widgets. Using various examples, this talk will help you understand what jQuery can do so you can see if it's right for your next project.

JavaScript Beyond the Basics by Nathaniel Schutta

JavaScript is one of the most widely used languages around and yet its also one of the most misunderstood. With Ajaxified UIs becoming the norm, this humble language is once again at the forefront.

Agile UI by Nathaniel Schutta

Some developers assume that agility and usability are mutually exclusive - in reality, they are extremely complimentary; if you squint, you might have a hard time telling the difference between agile practices and good user interface design. This usability talk is aimed squarely at developers giving you the tools you need to develop UIs that won't make your users yack. We'll discuss the importance of observation, personas, paper prototyping, usability testing and the importance of good moderators. In addition, we'll map the various aspects of user interface design to a typical agile iteration.

Hacking Your Brain for Fun and Profit by Nathaniel Schutta

The single most important tool in any developers toolbox isn't a fancy IDE or some spiffy new language - it's our brain. Despite ever faster processors with multiple cores and expanding amounts of RAM, we haven't yet created a computer to rival the ultra lightweight one we carry around in our skulls - in this session we'll learn how to make the most of it. We'll talk about why multitasking is a myth, the difference between the left and the right side of your brain, the importance of flow and why exercise is good for more than just your waist line.

Performance and Scalability Revisited: In-Memory Data Grids by Aleksandar Seovic

Building scalable, highly-available applications that perform well is not an easy task. These features cannot be simply "bolted" onto an existing application - they have to be architected into it. Unfortunately, the things we need to do to achieve them are often in conflict with each other, and finding the right balance is crucial.

In-Memory Data Grids: Not Your Mom's Cache by Aleksandar Seovic

While many developers still think of in-memory data grids as clustered caches, in reality they are much more and provide a solid foundation for the next generation of scalable web and enterprise applications.

Spring 3 into REST by Ken Sipe

REST as an architectural approach is greatly simplified through the selection of framework or tool to help with the tedious and repetitive template style that it demands. Until recently, some of the best approaches were through frameworks that required the use of languages other than Java, such as Rails or Grails. In the Java space, the choices were limited. The newly released Spring 3 changes that. One of the most significant changes in Spring 3 is its support for REST, which includes client as well as server support. **Prerequisite:** *Java 5*

Enter The Gradle by Ken Sipe

First in the Java Build space there was ANT, and there was a reliable way to build without an IDE. Then there was Maven, which provided standardization in build life-cycles and dependency management. Now... Enter the Gradle, which provides convention over configuration approach to the build process and an approach at building that isn't based XML. **Prerequisite:** *Some Groovy helpful*

Debugging your Production JVM by Ken Sipe

So your server is having issues? memory? Connections? Limited response? Is the first solution to bounce the server? Perhaps change some VM flags or add some logging? In today's Java 6 world, with its superior runtime monitoring and management capabilities the reasons to bounce the server have been greatly reduced.

XSS-Proof by Ken Sipe

Companies have focused for years to solidify the back-end infrastructure in defense against hacking attempts. Most companies however are forced to open up many ports including port 80 (http) for users to access web applications among other resources. This has led to web attacks growing to be the #1 classification of hacker attacks today. In this space Cross Site Scripting (XSS) is the #1 ranked vulnerability affecting a large number of sites. This evolution requires that the understanding of securing an application move beyond sys admins and incorporate all aspects of system delivery for the protection of a system and system resources.

HTML 5 ... and the Kitchen Sink by Brian Sletten

HTML 5 is an adventurous and confusing prospect that will help change the Web as we know it. It is being finalized as a standard but won't be fully supported by most browsers for quite some time. Companies like Apple and Google have already committed to it as the future of Web application development, however. There are a huge number of new features, updates and gotchas coming at us (including the proverbial kitchen sink!) so it is time to get prepared. This talk will walk you through the new bits and try to put it all into perspective.

REST : Information-Driven Architectures for the 21st Century by Brian Sletten

There is a shift going on in the Enterprise. While still used and useful, the promises of the SOAP/WSDL/UDDI Service-Oriented Architecture (SOA) stack have failed to live up to their promise. A new vision of linked information is enveloping online and Enterprise users. The REST architectural style is squarely behind this thinking as a way of achieving low-cost, flexible integration, increased data security, greater scalability and long-term migration strategies. If you have dismissed REST as a toy or are unfamiliar with it, you owe it to yourself to see what is so interesting about this way of doing things.

RDFA : Weaving Richness and Meaning in the Web by Brian Sletten

The human web is reasonably well in hand by now. We are getting pretty good at building systems that people find valuable and entertaining. We have not spent as much time concerned about our software friends. There is a ton of rich content available on the web that is too difficult to extract in automated ways using just XHTML, the meta tag and microformats. This talk will introduce you to some emerging technologies from the Semantic Web camp to enrich your web pages with useful information for both automated extraction and improved browsing experiences.

SPARQL : Querying the Web of Data by Brian Sletten

The human-friendly Web is about nicely-formatted, accessible content for users to browse. There are emerging Data Webs (both public and private) that rely on technologies from the Semantic Web stack to link increasingly rich connections between various data sources. SPARQL and RDF are the main tools for expressing and using this connectivity. This talk will introduce you to one of these topics and the practical and accessible aspects of employing them on the Web and in the Enterprise. Getting people to come to consensus on common models and schemas is usually the hardest part of any data integration strategies. These technologies help lower the bar on both the technical and social costs of stepping up your integration strategies.

Semantic SOA : Meaningful Service Strategies by Brian Sletten

The goal for web services was always to reduce our burden by increasing the potential for reuse of business functionality. Somehow, we got lost along the way in a morass of confusing, unfulfilling and downright broken technologies. While we are interested in pursuing REST-based systems for managing information, we need some strategies for tying it all together sensibly. If we abandon WSDL, SOAP and UDDI, what do we replace them with? This talk will walk you through combining resource-oriented strategies with technologies from the Semantic Web to describe, find, and bind to services in dynamic, flexible and extensible ways. We will start to blur the distinction between data, documents, services and focus on information and how it is

connected to what we already know. **Prerequisite:** *The Semantic Web: The Future Now, Give it a REST and SPARQL : Querying the Data Web would all be helpful talks to have attended*

What's Brewing in Java by Venkat Subramaniam

Java has come a long way, and yet there is so much that's happening in this space. In this presentation we will take a look at the exciting additions and changes coming up in the next version of Java. **Prerequisite:** *Good programming knowledge of Java*

How to Approach Refactoring by Venkat Subramaniam

You can't be agile if your code sucks. You know that you have to constantly refactor your code and design. But the question is how? In this presentation, instead of looking at a laundry list of refactoring techniques, we will instead look at how to effectively approach refactoring and along the way discuss some core principles to look for.

Programming Scala by Venkat Subramaniam

Scala is a static fully object-oriented, functional language on the JVM. While taking advantage of the functional aspects, you can continue to make full use of the powerful JVM and Java libraries.

Scala Tricks by Venkat Subramaniam

Scala is a very powerful hybrid functional pure object oriented language on the JVM. Scala is known for its conciseness and expressiveness. In this presentation we will look at some common tasks you do everyday in developing applications and see how they manifest in Scala.

Testing with dependencies by Venkat Subramaniam

Testing is a key ingredient to the success of a project. However, testing becomes awfully hard when your application deals with dependencies and that is often the reality.

Transforming to Groovy by Venkat Subramaniam

Groovy is a elegant, dynamic, agile, OO language. I like to program in Groovy because it is fun and the code is concise and highly expressive. Writing code in a language is hardly about using its syntax, however. It is about using the right idioms. Come to this section to pick up some nice Groovy idioms. **Prerequisite:** *Some knowledge of Groovy is helpful but not required.*