

# Lone Star Software Symposium: Austin

Marriott Austin Airport Hotel

July 10 - 12, 2009

<http://www.nofluffjuststuff.com/conference/austin/2009/07/home>

Fri, Jul. 10, 2009				
	1	2	3	4
12:00 - 1:00 PM	REGISTRATION			
1:00 - 1:15 PM	WELCOME			
1:15 - 2:45 PM	The Busy Java Developer's Guide to Java7 Ted Neward	Real-world Refactoring Neal Ford	JSF 2.0: An Introduction David Geary	Domain Driven Design and Development In Practice - Code Generation Srin Penchikala
2:45 - 3:15 PM	BREAK			
3:15 - 4:45 PM	The Busy Java Developer's Guide to Java Platform Security Ted Neward	Test Driven Design Neal Ford	JSF 2.0: Advanced Topics David Geary	Architecture Rules Enforcement using Aspects Srin Penchikala
4:45 - 5:00 PM	BREAK			
5:00 - 6:30 PM	The Busy Java Developer's Guide to Advanced Platform Security Ted Neward	The Productive Programmer: Mechanics Neal Ford	Flex for Java Developers David Geary	Domain Event Driven Architectures using Aspects and Event Stream Processing Srin Penchikala
6:30 - 7:15 PM	DINNER			
7:15 - 8:00 PM	Keynote: Venkat Subramaniam			

Sat, Jul. 11, 2009				
	1	2	3	4
8:00 - 9:00 AM	BREAKFAST			
9:00 - 10:30 AM	Emergent Design & Evolutionary Architecture Neal Ford	Spring 3.0 Overview Scott Leberknight	GWT: An Introduction David Geary	Programming Scala Venkat Subramaniam
10:30 - 11:00 AM	BREAK			
11:00 - 12:30 PM	The Busy Java Developer's Guide to Collections Ted Neward	Groovier Spring (More Flexible Applications With Spring and Groovy) Scott Leberknight	GWT: Advanced Topics David Geary	Effective Java Venkat Subramaniam
12:30 - 1:30 PM	LUNCH			
1:30 - 3:00 PM	Open Source Debugging Tools, A 90 Minute Bootcamp Matthew McCullough	Designing for Ajax, part 1 Nathaniel Schutta	Real World Hibernate Tips (Reloaded) Scott Leberknight	Cleaning up Code Smell Venkat Subramaniam
3:00 - 3:15 PM	BREAK			
3:15 - 4:45 PM	Git Going with Distributed Version Control Matthew McCullough	Designing for Ajax, part 2 Nathaniel Schutta	Polyglot Persistence Scott Leberknight	Building External DSLs Venkat Subramaniam
4:45 - 5:45 PM	BIRDS OF A FEATHER SESSION			

Sun, Jul. 12, 2009				
	1	2	3	4
8:00 - 9:00 AM	BREAKFAST			
9:00 - 10:30 AM	Architecting Code for Concurrent Execution: Theory and Practice Robert Fischer	REST : Information-Driven Architectures for the 21st Century Brian Sletten	iZero: Starting Projects Right Stuart Halloway	JavaScript: the Good, the Bad, and the Ugly Nathaniel Schutta
10:30 - 11:00 AM	MORNING BREAK			
11:00 - 12:30 PM	The Concurrency Toolset: JConch, Google Collections, and java.util.concurrent Robert Fischer	RESTlet for the Weary Brian Sletten	Taking Agile From Tactics to Strategy Stuart Halloway	Mastering Maven 2.0 Matthew McCullough
12:30 - 1:15 PM	LUNCH			
1:15 - 2:15 PM	EXPERT PANEL DISCUSSION			
2:15 - 3:45 PM	A Practical Take on GORM Robert Fischer	SPARQL: Querying the Data Web Brian Sletten	Hacking Your Brain for Fun and Profit Nathaniel Schutta	Java.next #1: Common Ground Stuart Halloway
3:45 - 4:00 PM	BREAK			
4:00 - 5:30 PM	Grails for the Enterprise Robert Fischer	Semantic SOA : Meaningful Service Strategies Brian Sletten	iPhone Objective-C integration to Java Web Services Matthew McCullough	Programming Clojure Stuart Halloway

# Lone Star Software Symposium: Austin

## -Session Schedule-

(event schedule as of July 8, 2009)

### Friday, Jul. 10

12:00 - 1:00 PM : REGISTRATION

1:00 - 1:15 PM : WELCOME

1:15 - 2:45 PM - Sessions

#### **Session #1 : The Busy Java Developer's Guide to Java7 by Ted Neward**

Even though the Java 7 JSR has yet to be formed, some interesting things are beginning to emerge from Sun about what Java7 may include when its formal release contents are finally made public.

#### **Session #2 : Real-world Refactoring by Neal Ford**

Refactoring is a fine academic exercise in the perfect world, but we don't really live there. Even with the best intentions, projects build up technical debt and cruddy bad things. This session covers refactoring in the real world, at both the atomic level (how to refactor towards composed method and the single level of abstraction principle) to larger project strategies for multi-day refactoring efforts. This talk provides practical strategies for real projects to effectively refactor your code.

#### **Session #3 : JSF 2.0: An Introduction by David Geary**

This session introduces JSF 2.0 fundamentals, with emphasis on new features in JSF 2.0. *Prerequisite: Familiarity with JSF, or other component-based frameworks*

#### **Session #4 : Domain Driven Design and Development In Practice - Code Generation by Srin Penchikala**

Domain Driven Design (DDD) technique helps architects and developers in mapping business domain concepts to software artifacts. When applying DDD techniques in a Java application, there are several domain classes and other artifacts (like configuration files) in Domain, Service and Application architecture layers that can be automatically generated using Code Generation techniques. For example, in a typical web application same data flows through the application layers in different forms like Domain Object (DO), Data Transfer Object (DTO) and XML. The developers usually write these classes manually from scratch or write the first class and then copy and paste when they need the classes for other use cases. These and other classes like Facade, Data Access Object (DAO interface, implementation and the unit test classes) are good candidates for code generation. Code generation helps the software development process by generating consistent code for the classes and other artifacts that have infrastructure (boiler-plate) logic but don't have any business logic. It fits in very well with Domain Driven Design philosophy of "Domain First, Infrastructure Second" where the developers can focus on business domain concerns and take the advantage of code generation to address the infrastructure concerns. If Domain Driven Design provides the "What", it is the code generation, as a development technique, that can provide the "How". Code generation is a good long-term solution even though this effort involves some initial investment (in terms of coding and time) which pays itself off in medium to large domain models. This technical session will discuss the role of code generation in a Java web application from the Domain Driven Design and Development context. I will start off the presentation with a quick overview of the Layered Architecture in a typical web application that was developed using DDD design pattern and implemented using Spring Framework (for middleware services) and JPA (with Hibernate3) for persistence. I will talk about the code generation concept in the project, that starts with the creation of a standard and consistent project template (using Maven Archetype feature). The discussion will include a list of the classes we need to write from scratch (like XSD, Domain Object and Service) and then code generate the Java classes and configuration files in domain and other layers (like DAO's, Factories, Repositories, Delegates, DTO's, Unit Tests, and Spring configuration files).

2:45 - 3:15 PM : BREAK

3:15 - 4:45 PM - Sessions

#### **Session #5 : 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.

#### **Session #6 : 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.

#### **Session #7 : JSF 2.0: Advanced Topics by David Geary**

This session covers two of the most kickass features of JSF 2.0: composite components and built-in Ajax. *Prerequisite: Familiarity with JSF, or other component-based frameworks. Familiarity with Ajax. This session builds on the JSF 2.0 Introduction talk, so it is helpful, although not required, if you attend the intro talk before coming to this session.*

# Lone Star Software Symposium: Austin

## -Session Schedule-

(event schedule as of July 8, 2009)

### Session #8 : Architecture Rules Enforcement using Aspects by Srinu Penchikala

Even though many companies have some kind of application architecture standards, they don't usually have a mechanism to enforce those standards. As a result of this lack of architecture governance, the Implementation (Code) often doesn't match the Requirements (Reference Architecture). Enforcing Reference Architecture guidelines promotes consistency and modularity in the System. It also helps in detecting structural complexity and preventing it earlier in the software development process. As a result, the application code is modifiable, portable, and testable. In this presentation, I will talk about the significance of enforcing the architecture rules and standards and how to actually enforce them in software development projects. I will start off the discussion explaining various types of architecture rules that range from simple module dependencies to enforcing how APIs and specific idioms should be used. All these rules usually fall into categories such as Layered Architecture, Separation Of Concerns, Domain Driven Design, and Infrastructure. Some of the rules examples will include basic validations like DAO (Persistence) layer shouldn't depend on any other layers, Presentation layer should not use DAO classes directly, and Service layer should never call web layer. Separation of concerns rules are no transaction management in DAO classes and Service layer should be transactional. More sophisticated rules like that are used in a Domain Driven Design project such as Service object creation through a Factory object (i.e. no Service instantiation using the new keyword), No direct access to DAO's except from the Domain classes (go through Domain or Repository objects), and Business service that fails with a concurrency related failure can be retried. **Prerequisite:** *This is an advanced AOP talk. It assumes the attendees are familiar with AOP and Aspects design concepts.*

4:45 - 5:00 PM : BREAK

5:00 - 6:30 PM - Sessions

### Session #9 : The Busy Java Developer's Guide to Advanced Platform Security by Ted Neward

So you know the platform security model, and now you want to use it in new and interesting ways, like creating a custom Policy implementation, a custom Permission, or create a custom security context in which code will execute. Perhaps you even wish to make certain objects accessible only to those with the right permissions, or cryptographic key. Nothing could be easier, despite Java security's reputation as a dark and arcane place. **Prerequisite:** *The Busy Java Developer's Guide to Platform Security*

### Session #10 : The Productive Programmer: Mechanics 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!

### Session #11 : Flex for Java Developers by David Geary

An introduction to Flex for Java developers. **Prerequisite:** *Familiarity with Flex and at least one other web application framework*

### Session #12 : Domain Event Driven Architectures using Aspects and Event Stream Processing by Srinu Penchikala

This presentation will give an overview of using Aspects and AOP to implement business domain event processing logic and Event Stream Processing (ESP) to analyze event data in a real-time. The presentation uses a sample application to demonstrate the use of ESP in a domain application. The sample application uses technologies like Aspects, Spring AOP, JMS (ActiveMQ), JMX, and ESP (Esper) to manage the business domain events. **Prerequisite:** *Attendees should have some familiarity with Aspect-oriented Programming (AOP) concepts.*

6:30 - 7:15 PM : DINNER

Keynote: \$presentation.title - Venkat Subramaniam

## Saturday, Jul. 11

8:00 - 9:00 AM : BREAKFAST

9:00 - 10:30 AM - Sessions

### Session #13 : Emergent Design & Evolutionary Architecture by Neal Ford

Most of the software world has realized that BDUF (Big Design Up Front) doesn't work well in software. But lots of developers struggle with this notion when it applies to architecture and design. Surely you can't just start coding, right? You need some level of understanding before you can start work. This session describes the current thinking about emergent design & evolutionary architecture, including both proactive (test-driven development) and reactive (refactoring, composed method) approaches to discovering design. The goal of this talk is to provide nomenclature, strategies, and techniques for allowing design to emerge from projects as they proceed, keeping you code in sync with the problem domain.

### Session #14 : Spring 3.0 Overview by Scott Leberknight

The Spring framework has simplified Java enterprise and web development since 2003, and has been a major innovator in improving and simplifying Java server-side programming since its inception. This session will look at the new features in Spring 3.0 as well as what's being removed from the Spring core.

# Lone Star Software Symposium: Austin

## -Session Schedule-

(event schedule as of July 8, 2009)

### **Session #15 : GWT: An Introduction by David Geary**

An introduction to Google Web Toolkit. *Prerequisite: Familiarity with a component-based framework, preferably a desktop application framework*

### **Session #16 : 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.

10:30 - 11:00 AM : BREAK

11:00 - 12:30 PM - Sessions

### **Session #17 : 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.

### **Session #18 : Groovier Spring (More Flexible Applications With Spring and Groovy) by Scott Leberknight**

Spring provides a solid foundation for web and enterprise applications. Its support for dynamic languages like Groovy adds interesting capabilities that can make your application architecture more flexible and dynamic.

### **Session #19 : GWT: Advanced Topics by David Geary**

Learn to do really cool stuff with GWT. *Prerequisite: The GWT: Introduction session is not a prerequisite for this session, but it will help if you have some familiarity with GWT.*

### **Session #20 : Effective Java by Venkat Subramaniam**

Java is a well established language, that has been around for more than a decade. Yet, programming on it has its challenges. There are concepts and features that are tricky. When you run into those, the compiler is not there to help you.

12:30 - 1:30 PM : LUNCH

1:30 - 3:00 PM - Sessions

### **Session #21 : Open Source Debugging Tools, A 90 Minute Bootcamp by Matthew McCullough**

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. This session will quickly survey a wide range of tools across the Java, Networking, Filesystem, SOAP, REST, HTML, CSS and JavaScript realms. We'll look at applications such as VisualVM, which help you analyze your heap and garbage collection cycles of both local and remote applications. Performance and load testing tools such as JMeter will expose bottlenecks, threading, and scalability concerns of everything from Java modules to Web Apps (even ones that don't use any Java). *Prerequisite: Basic knowledge of web services, core Java, web application design.*

### **Session #22 : Designing for Ajax, part 1 by Nathaniel Schutta**

So you've convinced the boss that your new web application just has to have Ajax...but now what? With dozens of libraries making even the most blinkish of interactions trivial, how do you decide where to sprinkle the magic Ajax dust? This talk will give a plain old boring "web 1.0" an Ajax facelift with a focus on improving the user experience providing you with a game plan for introducing Ajax to your world.

### **Session #23 : Real World Hibernate Tips (Reloaded) by Scott Leberknight**

Hibernate is a very powerful object/relational mapping framework. This session contains a new set of Hibernate tips, tricks, and pitfalls.

### **Session #24 : Cleaning up Code Smell by Venkat Subramaniam**

Projects often start out simple, but soon become complex and turn into a lose cannon. Organizations are struggling to maintain and evolve software. Poor code quality is a significant part of that problem. Improving the quality of code is critical to success of enterprise projects.

3:00 - 3:15 PM : BREAK

3:15 - 4:45 PM - Sessions

### **Session #25 : Git Going with Distributed Version Control by Matthew McCullough**

Many development shops have made the leap from RCS, Perforce, ClearCase, PVCS, CVS, BitKeeper or SourceSafe to the modern Subversion (SVN) version control system. But why not take the next massive stride in productivity and get on board with Git, a distributed version control system (DVCS). Jump ahead of the masses staying on Subversion, and increase your team's productivity,

# Lone Star Software Symposium: Austin

## -Session Schedule-

(event schedule as of July 8, 2009)

debugging effectiveness, flexibility in cutting releases, and repository redundancy at \$0 cost. Understand how distributed version control systems are game-changers and pick up the lingo that will become standard in the next few years. **Prerequisite:** *Basic understanding of Subversion or similar version control system*

### **Session #26 : Designing for Ajax, part 2 by Nathaniel Schutta**

We'll pick up where Part 1 left off working in even more advanced approaches such as offline support with Google Gears.

### **Session #27 : Polyglot Persistence by Scott Leberknight**

Polyglot persistence is all about considering your persistence requirements and selecting a persistence mechanism that best meets those requirements, as opposed to selecting an RDBMS as the default choice. In this session we'll look at some of the persistence alternatives that are available like Amazon SimpleDB, CouchDB, Google Bigtable, and more.

### **Session #28 : Building External DSLs by Venkat Subramaniam**

Domain Specific Languages (DSLs) are languages targeted at a particular problem and domain. They have context and are fluent. They help users of applications at various levels to easily communicate with your application. Developing DSLs, however, are not easy. You could easily get dragged into using parsers and tools with steep learning curve.

4:45 - 5:45 PM : BIRDS OF A FEATHER SESSION

## **Sunday, Jul. 12**

8:00 - 9:00 AM : BREAKFAST

9:00 - 10:30 AM - Sessions

### **Session #29 : Architecting Code for Concurrent Execution: Theory and Practice by Robert Fischer**

The power of multicore machines and cloud computing is all dependent on an application's ability to successfully leverage concurrency. Although concurrency has traditionally been considered fatally difficult in Java, a few simple architecture principles can make all the difference. This session will review some of those principles in both theory and practice.

### **Session #30 : 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.

### **Session #31 : IZero: Starting Projects Right by Stuart Halloway**

If an iteration is the heartbeat of an agile development process, then Iteration Zero (IZero) creates the heart. While you can (and should) retrospect and adjust throughout the software lifecycle, few things are as valuable as a good start. In this talk, you will learn how we run Iteration Zero at Relevance.

### **Session #32 : JavaScript: the Good, the Bad, and the Ugly by Nathaniel Schutta**

Thanks to Ajax, JavaScript is cool again and developers are taking a second look at this much maligned language. This session will give you an overview of this misunderstood language as well as opening your eyes to some of the excellent tools available to ease the pain of developing in this dynamic language.

10:30 - 11:00 AM : MORNING BREAK

11:00 - 12:30 PM - Sessions

### **Session #33 : The Concurrency Toolset: JConch, Google Collections, and java.util.concurrent by Robert Fischer**

JConch is a library that provides a few high-level tools for high-concurrency environments on the JVM. The java.util.concurrent package in the Java standard library provides low-level structures for managing concurrent communication. Learn here how to use both of them in order to produce clean, highly-concurrent, and highly-tunable code.

### **Session #34 : 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. **Prerequisite:** *REST (unless you are very comfortable with REST)*

# Lone Star Software Symposium: Austin

## -Session Schedule-

(event schedule as of July 8, 2009)

### **Session #35 : Taking Agile From Tactics to Strategy by Stuart Halloway**

Teams adopting agile should begin at a tactical level, but they shouldn't end there. The Agile Manifesto operates at many different levels. Learn to apply the principles of agile at a strategic level. Otherwise you can have a great agile ground game and still lose.

### **Session #36 : Mastering Maven 2.0 by Matthew McCullough**

Maven has been on the Java build tools scene for quite a number of years, but the adoption rate in enterprises is now going through the roof. Maven can seem daunting, but this presentation will equip existing Maven users with more efficient techniques and tools to overcome the biggest perceived Maven hurdles and build issues with ease. We'll examine tools to help you find artifacts in central repositories, manage your corporation's internal Maven artifacts with a proxy tool such as Nexus, view and override dependency graphs, dependency management and multi-module best practices, create OS specific profiles, and leverage the latest Maven plugins for the top Java IDEs. **Prerequisite:** *Basic Maven knowledge*

12:30 - 1:15 PM : LUNCH

1:15 - 2:15 PM : EXPERT PANEL DISCUSSION

2:15 - 3:45 PM - Sessions

### **Session #37 : A Practical Take on GORM by Robert Fischer**

For years, the venerable Hibernate object-relational mapping framework has dominated the persistence scene in Java. The Grails web application framework extended Hibernate and Spring with their impressive GORM persistence framework, providing convention-over-configuration development to the O/RM and DAO layers. This session will introduce GORM and work up to some of its intermediate features.

### **Session #38 : SPARQL: Querying the Data Web by Brian Sletten**

The human-friendly Web is about nicely-formatted, accessible content for users to browse. There is an emerging Data Web that relies 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 the practical and accessible aspects of employing these ideas on the Web and in the Enterprise. **Prerequisite:** *The Semantic Web: The Future, Now and Rich Web Pages : Publishing Semantic Content with GRDDL and RDFa would both be helpful but are not required*

### **Session #39 : 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.

### **Session #40 : Java.next #1: Common Ground by Stuart Halloway**

In this talk, we will explore and compare four of the most interesting new JVM languages: Clojure, Groovy, JRuby, and Scala. Each of these languages aims to greatly simplify writing code for the JVM, and all of them succeed in this mission. However, these languages have very different design goals. We will explore these differences, and help you decide when and where these languages might fit into your development toolkit. For more information see <http://blog.thinkrelevance.com/2008/8/4/java-next-common-ground>.

3:45 - 4:00 PM : BREAK

4:00 - 5:30 PM - Sessions

### **Session #41 : Grails for the Enterprise by Robert Fischer**

The Grails web application is an innovative hybrid of best-of-breed Java technologies and dynamic/convention-based development. The result is a powerful, flexible, exciting framework that still fits comfortably into enterprise stacks. This session introduces Grails, but approaches it from the perspective of an enterprise web development stack, in order to see how Grails works well in mid-size and mature development shops.

### **Session #42 : 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. **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*

### **Session #43 : iPhone Objective-C integration to Java Web Services by Matthew McCullough**

iPhone development is all the rage both in the mobile entertainment, social networking, and productivity application spaces. As a Java developer, prepare yourself to be a participant in aspects of this new breed and platform of development. Hop on board with a quick

# Lone Star Software Symposium: Austin

## -Session Schedule-

(event schedule as of July 8, 2009)

start to iPhone application coding in Objective-C and integration with some of our favorite Java web service back-ends such as JSR311 Jersey and RESTful Grails.

### **Session #44 : Programming Clojure by Stuart Halloway**

Find out why Clojure is Java.next: \* Clojure provides clean, fast access to all Java libraries. \* Clojure provides all the low-ceremony goodness you know and love from dynamic languages such as Ruby and Python. \* Clojure includes Lisp's signature feature: Treating code as data through macros. \* Clojure's emphasis on immutability and support for software transactional memory make it a viable option for taking advantage of massively parallel hardware.