

## Overview

Writing enterprise server software leveraging Java (RxJava), Scala, Groovy, SQL, Hadoop, VoltDB, Neo4j, Bash+Fish, Spring, Finagle, Google Protocol Buffers, OSGi, Mockito, JUnit, YAML, sometimes front-end stuff, is my daily pursuit. I know of but have not used Swift, Coffee Script, HTTP/2, and others. I've done some Android and iOS coding. I'm pretty strong with Unix, Linux, tmux, Amazon Web Service, Puppet, and Zookeeper, Jetty, Tomcat, J-Boss. I like IntelliJ IDEA, and Atlassian tools. Apple OS-X with Homebrew, or MacPorts, possibly Fink is good.

Employment-wise, I worked at Sun Microsystems within software engineering and as a Java Consultant to their clients teaching architecture, methodology, and design and coding (RUP, UP, SunTone, Agile, patterns). I later worked at RedPrairie/JDA Software Group as a Senior Engineer doing OSGi SOA and leading one new product teams. I was a Senior Architect at Packexpo, Inc.. I plan to continue this stuff, writing and operating beautiful code, collaborating realizing business value through software.

## Stories

### *RxJava*

With “Rx (Reactive),” “back pressure” manages load to ensure service level requirements. At the extreme, the fixed length request queue and finite number of processing threads simply deny excess requests, but crucially also producer throttling, consumer block size setting, and rate adjusting balance resource consumption. Some of these may automatically adjust themselves. Reactive defines the standards enabling cross-language compatibility. “Reactive [systems are] Responsive, Resilient, Elastic, and Message-Driven.”<sup>1</sup> Future transforms and callbacks enable response code and flatmap operations collapse nested Futures. This is super cool stuff, but Reactive Streams is where it becomes codable:

“Reactive Streams is an initiative to provide a standard for asynchronous stream processing with non-blocking back pressure.”<sup>2</sup> RxJava v1.0 predated reactive streams, but an adapter is available. Streams may be processed, split, and composed.

I used RxJava within server-side service code and around in inter-service calls on one item trading website.

### *Finagle*

Like Netflix's RxJava, Twitter's OSS Finagle enables asynchronous (reactive) programming on the JVM. It predated RxJava.

I used Finagle around inter-service calls on one application, but with the absence of stream processing, callback hell was endured, massively decreasing code readability, thereby maintainability.

---

1 <http://www.reactivemanifesto.org/>

2 <http://www.reactive-streams.org/>

## ***Google Protocol Buffers***

Protocol Buffers provides fast, reliable remote communications. Definitions may be made backwards compatible to the extent that server-side may advance ahead of client-side version.

I used protobuf for inter-service communications on one project.

## ***Puppet***

Puppet facilitates cloud environment service, presence and configuration.

I used Puppet at two companies and investigated at JDA.

## ***Zookeeper***

Zookeeper provides strict consistency across nodes often used for locking/synchronization within clusters of processing nodes.

I used Zookeeper a couple of companies and investigated it for use with OSGi at JDA.

## ***Nagios***

Nagios is OSS and facilitates system and service monitoring. I have used it.

## ***OSGi***

OSGi manages versioned components and classpaths at runtime with strict isolation, also the location of call targets, be they local or remote, is transparent to the caller.

I designed and coded our first uses of it at JDA. Apache Karaf and Felix were our choices.

## ***CXF***

CXF provides service exposure and connection solely through configuration. It enables SOAP, REST, and RMI to be configured as endpoints, even concurrently active ones.

I prototyped CXF for use within OSGi at JDA.

## ***Objective-C iOS***

Before Swift, iOS direct native apps were coded in Objective-C.

I was trained in iOS coding in Xcode and JetBrains' ReSharper at JDA as a developer.

## ***Android***

Android, as a variant of Linux and its programming similar to Java, it is highly accessible.

I trained myself on it and did a little prototyping of one iOS app I had written.

## ***Mockito***

Mockito often beats EasyMock. Mocking, of course, is crucial for facilitating TDD, to make the unit tests work. As BDD with Cucumber is totally awesome, Mockito, JUnit, Cucumber are sorely missed when not present.

I've used Mockito and PowerMock to spice it up a bit.

## ***Cucumber***

Cucumber enables acceptance tests to be written and maintained by business stakeholders and they compile into actual test code. Pretty cool.

I used it at one company.

## ***Atlassian Tools***

Atlassian provides rich interoperable tool suites from Git clients to chatting clients to issue tracking, code reviews, wiki's, and continuous integration servers.

I used most of Atlassian's tools at Tendril and JDA as well as their Intelli-J IDEA plugin with which pre-commit Crucible code review creation was facilitated.

## ***Groovy***

Groovy is a meta programming tool facilitating domain specific language (DSL) creation. Many tools and frameworks, Gradle being one for example, are Groovy DSL's. Groovy brings closures and other new language features to the JVM. It may be ideal for rapid prototyping. Initially it was 1000x slower than Java and contained serious memory leaks, but it's pretty good now, merely 10% slower or so.

I've used it for shell scripting, although I like Fish and tolerate Bash, and REPL code testing and experimenting, sometimes for unit tests.

## ***Grails***

Similar to Rails in Ruby, Grails in Groovy uses convention over configuration producing web GUI's pretty tersely. It generates the scaffolding, similar to Spring Roo.

I used grails to create an RDB, web CRUD prototype at JDA proving that SQL Server (relational) was

faster than Neo4j (non-relational/nosql) for our generic system. Personally I also created a trading web site using Monet DB which enables users to post items with images and descriptions which they desired to trade. Users would matchup trades.

## ***Scala***

Scala is object oriented functional language with map, flatmap, for-comprehension, and other built-in tools which facilitate reactive programming. It is impressively readable ways, and being functional TDD and unit testing slightly reduces and alters in necessity.

I completed Coursera's Scala training worked with Scala code at Tendril as a developer.

## ***JEE***

EE collects many technologies used in enterprise computing, JTA, JDA, Entity Beans, Session Bean, Message Driven Beans, JMS, Servlets, JSP, JSTL, etc.. Recent versions enabled individual components to be used individually and embedded, outside of an application server.

A decade ago I was trained in the precursors and in J2EE at Sun Microsystems and I designed systems for their customers using it while training their customers in it. I continued using it elsewhere, usually in JBoss, but also in WebLogic. I know of recent versions, but primarily have at work used Jetty and Tomcat alone, often reverse-proxied by Apache or Nginx.

## ***JBoss***

An OSS JEE application server and related suites of products.

I have used JBoss professionally since version 2.x. Sun customers, Packexpo, Etilize, RedPrairie/JDA, and others using JMS/MoM, jBPM and merely as a finite state machine (FSM).

## ***Jetty, Tomcat***

A web container with native components which reduce CPU load during image transfer among other things.

I have used them at almost every employer, including JBoss Web, a variant of Tomcat/Catalina.

## ***Unix, Linux FreeBSD/Darwin/OS-X***

BeOS and GNU Operating System attempted to be completely asynchronous and concurrent, a noble goal, but found correctness and debugging overly daunting challenges. The winners to date remain traditionally programmed variants of Unix which continuously and steadily evolve over time reaching new platforms, accommodating new and changing hardware, and debugging and improving their own code.

Obviously, I am a Unix guy including having worked at Sun, been a consultant representing Sun, coded in Linux in Operating Systems as an undergraduate, and studied system administration under Evi Nemeth (an originator of the "sudo" command, author of the Unix System Administration Handbook, among many other unique achievements) at the University of Colorado, Boulder.

## **Employment History**

**9/1/2014-2015 Software Engineer**, Comcast, Corp.

- Prototyped one of their systems against source data from a new source.
- Advanced one of their systems leveraging Reactive and quality coding principals.

**2/1/2014-5/1/2014 Software Engineer**, Tendril, Inc.

- Back-end dev-ops programmer on SOA Finagle+Protobuf+Cloud components.
- Technologies: Atlassian, Git, Cucumber BDD, PostgreSQL, Hadoop, Hibernate, Scala, Intelli-J IDEA.

**2/1/2010-12/31/2013 Senior Software Engineer**, JDA Software Group, Inc. which merged with RedPrairie, Corp. formerly StorePerform, Inc.

- Project Lead for integration of iOS, Android mobile task system with legacy web-based system
  - Tasks roughly are E-mail messages enhanced for distributing and tracking business actions.
  - Customers included Fortune-500 companies. Scalability, performance were significant.
  - Technologies: OSGi, Gradle, Git, and Sql Server.
- Extended and maintained legacy J2EE web-based task management system.

**2008-2009 Independent Software Development Consultant & Sun Java Consultant**

- Extended and maintained a JEE merchandise inventory system with MySQL DB.
- Customers: AmDocs, Etilize, Bell South, eBay, Frontier Airlines, Skyway

2007 Chief Operation Officer, Evolution Hosting, Inc.

- ViaWest data center Linux servers for J2EE applications with Java management software.
- Large number of customers including Crocks, Wells Fargo, and Hunter Douglass, among others.

2004-2007 Independent Software Development Consultant & Sun Java Consultant

- Created NIO client with could reach 100 Mb/s scale message rates.
- Created new web sites.

- Consulted at eBay, Frontier Airlines, Bell South, Skyway representing Sun Microsystems, Inc.

**2002-2004 Senior Software Architect**, Packexpo, Inc.

- Created a reporting systems for packaging advertising services.
- Maintained packaging advertising web site.
- Technologies: JBoss, Linux, MySQL, Intelli-J, BeanShell, JSP, Lzop'ed data stream for large DB backups over NFS mounts.

**1999-2002 Java Consultant**, Sun Java Center, Sun Microsystems, Inc.

- Represented Sun for RUP, UP, SunTone, OOA&D mentoring and J2EE coding.
- Consulted at Citigroup, Vanguard Fiduciary Trust, Credit Suisse.
- Performed an Architecture Assessment recommending SOA and JEE which later was followed as advised.

**1998-1999 Java Developer**, Sun Educational Services, Sun Microsystems, Inc.

- Created web-based and Swing computer assisted training systems.
- Improved the performance of our custom pseudo-recursive-descent parser template engine by adding caching of parser results and by altering the parsing algorithm.

## **Education, Certifications, Patents**

- Sun Certified Java Developer
- Sun Certified Java Programmer
- Patent: [#6907423](#) Search engine interface and method of controlling client searches
- B.S. Computer Science, University of Colorado at Boulder