Profile

Ben has over twenty years experience designing and developing software solutions within the investment banking and capital markets industry. He has a strong project track-record working with real-time trade processing systems and distributed messaging environments. Ben has specialized domain knowledge covering Credit, Fixed Income, Rates and Risk.

Career Resume

***Sr. IT Consultant, J.P.Morgan , New York, NY (Mar 2013 –Present)***

**Real-Time Liquidity Risk Technology.**

Designed, developed, deployed a distributed heterogeneous JCACHE data grid rendering/aggregation engine (Infinispan 5.3) onto a UHPC Linux supercomputer (240 CPU/3TB RAM/Infiniband OSI interconnect/ Oracle EXAdata physical data appliances ) to host JPMorgan Chase & Co. positions (40mm+ including retail, institutional, derivative, synthetics, exotics across entire LE balance sheet totalling $2T USD asset classes) facts, reference data, and computational dimension operands . Once deployed, data grid aggregation engine continuously consumed both interactive and batch quantitative/algorithmic/risk manager computational operations with the ambition to render and aggregate real-time on-demand liquidity risk to/from all stakeholder views (including U.S. regulators). Specific contributions included the build, configuration, and deployment of a Java 7 Sockets Direct Protocol empowered JVM set cluster that used native Infiniband RDMA to affect a ZERO-COPY transport (across IB verbs Fabric driver) from Oracle EXAdata physical appliance data sources \*directly\* to Java application NIO buffers (using only the java.net.Socket APIs) – 100% bypassing the Linux OSI network layers’ stack (physical/data/network/transport/session/application) – a 600% throughput improvement when compared to traditional TCP/IP/PPP/10gbEthernet network routes. Also contributed a POC research into the applicability of using Peter Lawrey’s openHFT HugeCollections solution (100% off-heap, 100% GC agnostic) as a JSR-107 compliant Cache implementation that would be 100% interoperable with JCACHE provider = Infinispan 5.3 (via its DataContainer adapter API). Designed numerous performance tests wrt to specific SLA expectations/obligations covering Scenario Manager STRESS on position overlay analyzers, available collateral, cash flow generators, sources and uses, and standard regulatory liquidity statements. Elected J.P.Morgan technical representative to the Java Community Process expert group (JSR-347) tasked with establishing the Java API standard for Distributed Data Grid technology.

***Sr. IT Consultant, Credit Suisse, New York, NY (Mar 2011 –Mar 2013)***

**Global Web Services. Capital Markets Technology.**

Elected Credit Suisse technical representative to the Java Community Process expert group (JSR-107) tasked with establishing the Java API standard for Caching technology. Duties including writing Chapter 5 of the JSR-107 DRAFT specification, detailing the standard requirements for Java caching providers with regards to Transaction processing. Designed the specific parts of the javax.cache.CacheBuilder API that provided mechanism for applications programmers to specify the Transaction isolation-level join point and view for any transactional JSR-107 CacheManager entity. Provided extensive documentation for both use case and reference implementation compatibility testing. Consulted with numerous JSR-107 CACHE provider vendors (Terracotta, EhCache, Gemfire, Oracle Coherence, JBoss Infinispan, Google Guava etc.) re: their products’ JSR-107 candidate builds passing the Test of Compatibilty suite covering both Hibernate L2 Cache SPI andTransactions. Senior member of Credit Suisse Technical Advisory Group architecting the Oracle Coherence JSR-107 implementation’s PaaS integration within CS global Java Application Platform. Consulted Credit Suisse Fixed Income eTrading technology teams re: impact of the new JSR-107 standard applicable to their architecting/building/deploying a best Caching solution join-point to their deployments. Delivered presentation to 200 members of NYC Java SIG titled “Data Locality, Latency and Caching: JSR107 and the new Java Standard”.

***Sr. IT Consultant, Citigroup, New York, NY (Mar ‘10 –Mar ‘11)***

**Credit Electronic Trading Systems**

Designed, developed, deployed, real-time, message-oriented, Java software components that integrated Citi Credit Trading desk front office sub-systems (Price Streaming Systems, Price Benchmark Systems, Reference Data Systems, RFQ & OMS Systems, Analytics, Books/Records/Positions Systems, and dozens of individual London/NY Trader Workstation GUI Blotters) with Citi institutional trading counterparties via industry leading ECNs (Bloomberg, Trade Web, Market-Axess, Citi Velocity). Software developed used the FIX 4.4 messaging protocol to facilitate the RFQ, QUOTE(ANSWERED QUOTE), and QUOTE-LIFT/NEW ORDER business use cases for the trading of core Credit products (Corp Bonds) and the trading of their derivatives (CDS, CDX, iTraxx). Gateway based STP communication/transport from these ECN facing front-office components (via ION MarketView/MMI connectivity bridge) to middle-office trading systems (Calypso) and Citi transactional systems of record (ledger/TPS). The centrally featured software component developed in the this credit eTrading process stack was a multi-threaded Java “Price Caching Service” (PCS) that facilitated the real-time caching, distribution, and event notification of Citi-internal streaming, customer-tiered, Price matrix updates. The PCS software component made extensive using of KX’s proprietary KDB+ data product, KX’s proprietary ‘q’ programming language, and Tibco EMS – designed in concert to ensure that the minimum mean Citi business latency mandate of a Price-update capability of > 1,000 updates per second was continuously met (and indeed exceeded).

**Environment**: Java 1.6, KDB+/q 2.5, Tibco EMS 5.x (JMS API), FIX 4.4, FpML 4.x, ION MarketView/MMI, RedHat Enterprise Linux, Sybase.

***Sr. IT Consultant, Lehman Brothers, New York, NY (Mar ‘09 – Mar ‘10)***

**Capital Markets IT**

Mortgage Capital Division, Aurora Bank FSB (non-bankrupt, non-liquidated FSB subsidiary of Lehman Brothers Holdings estate). Designed, developed, deployed message-oriented Java EE components (MDBs/POJOs)  that integrated Capital Markets Trading group’s middle office Asset-Liability Management, Risk Analytics, Books/Records, DTCC, Compliance and Settlement systems with the Bloomberg T/P OMS (Bloomberg AIM, Buy-Side).  Components fully satisfied Bloomberg gateway’s pub/sub/ack FIX protocol messaging requirements to trade GSE Agency MBS, MBS TBA, Interest Rate Swap, Interest Rate Swap Option (Swaption) and Tri-Party Repo asset classes via the Bloomberg OMS. Advised Capital Markets IT re: specific platform replacement recommendations (application, messaging, and data tiers) to best facilitate technology transition of services agreement w/ Barclays Capital re: legacy Lehman Brothers application platform.

**Environment**: Java SDK 1.6, Java EE 1.5, JBoss 5.x AS, Terracotta DSO API (HA, low-latency distributed cache), Sybase, Microsoft SQL Server, PostgreSQL, FIX protocol 4.x, QuickFIXj Engine, R/SAS OLAP MDX, LPS AFTgo.com Risk Modeling/Analytics, Apache Hadoop HDFS/MapReduce.

***Sr. IT Consultant, Merrill Lynch & Co., New York, NY (Mar ‘06 – Mar ‘09)***

**Municipal Trading Systems IT**

Designed, developed, and deployed core software components used by the Municipal Trading, Sales, Derivatives, and Underwriting desks (both front office and middle office). Core components implemented real-time pricing, trade capture, risk, trade/settlement confirmation, and transaction custody requirements for both institutional and retail STP specifications. Components technically implemented as lightweight C++ service processes, asynchronously integrated via Tibco RV, Sybase Open-Client/ct-lib persistence/transaction API , and deployed to an enterprise HA Sun SPARC/Solaris cluster. Designed and developed several Municipal Web Services deployed on a 100% open-source Java 1.4 EE technology stack. These Municipal web services deployed as SOAP endpoints via Apache Axis 1.4/Spring 2.5 MVC/Apache Tomcat 5.x HTTP Servlet Container. Java SOAP services integrated with the Core Trading System (C++) via the Tibco RV-proprietary Java API (non-JMS). Specific services developed and deployed include:

* A real-time Muni Position web service (specifically supporting these dynamically evolving municipal derivatives products – (a) ML-TRUST, (b) ML-RITE, and (c) ML-Synthetic Put-able Float). Web service is invoked via the firm-wide risk services framework, directly supporting mission-critical derivatives position requirements of the Municipal Arbitrage-Master/Trader Workstation project. (Java 1.4 EE, Apache Axis 1.4 SOAP, Spring 2.5, Apache Tomcat 5.x)
* A real-time “Quick Ladder” Muni Price update service that updates all ML trading channels’ existing retail offers in response to any municipal ladder pricing event initiated by the trading desk. (Solaris 5.9 SUNWSpro C++/RogueWave Source-Pro 3.0/Sybase 12.5 Open-Client Ct-lib, Tibco RV 7.4)
* A Muni Under-Lying Rating service (integrated with agency feeds from S&P, Moody and Fitch) empowering trading desks with the on-demand uninsured equivalent ratings value of any agency-rated municipal security. Same service also delivered on-demand re-insurer provider information for any agency-rated municipal security.
* An External Client Muni Offering Upload service, supporting both institutional (Bloomberg/MLX) and retail (TMC/SIG) client channels (Apache Axis 1.4 SOAP, JAX-RPC. Sybase jConnect JDBC, Tibco RV 7.4 Java API (non-JMS)).
* Enhanced municipal trading desk’s core Institutional Bid GUI application, supplementing presentation with extended data (Bloomberg subscribed) specific to the STP of Institutional clients’ Bid-Wanted requests, internal ML Bid-Wanted pricing, MLBloomberg Bid-Answer transmission, and subsequent Bloomberg initiated Bid-Answer Trade execution.

**Environment**: FIX protocol 4.x, Bloomberg XML/XSD, Motif/X11, Solaris C++, Sybase 12.5 Open-Client Ct-lib, Tibco RV 7.4

***Sr. IT Consultant, Goldman Sachs & Co., New York, NY (Oct ‘05 – Mar ‘06)***

**Fixed Income Currency and Commodities (FICC) Derivatives Technology**

Developed message-oriented Java components that implemented the Interest Rate Products group’s requirements for counter-party risk management, counter party trade confirmation, and counterparty contract settlement within the global Goldman FICC Derivatives Contract Business Model (CBM) IT framework. FICC derivatives products specifically supported include interest rate swaps (vanilla), interest rate swap options (swaptions), interest rate caps, etc. Components facilitated and maintained transactional custody of message information (FpML 4.1) during the straight-through-processing (STP) of IRP trades from front-office origination systems (Liberty), through the contract business model (Risk, Confirmations, Settlements) and on through to back end Goldman transactional systems of record (CSW/G.Ledger).

**Environment*:*** Java SDK 1.4, TIBCO EMS (JMS), BEA WebLogic, Sybase 12 RDBMS, Eclipse, proprietary GS DBMS (SecDB/Slang).

***Sr. IT Consultant, J.P. Morgan, New York, NY (Oct ‘02 – Oct ‘05)***

**Fixed Income Trading Division, Credit Derivatives Group**

Designed and developed secure J2EE intranet application (‘Spartak’) to implement the global trade capture facility for the investment bank’s credit derivatives straight-through-processing function (credit default swaps, interest rate swaps, total return swaps, credit spread options, credit default swap index (TRAC-X/iTraxx), and hybrid/exotic structured credit trades). ‘Spartak’ system implemented as a set of EJB 2.0 components deployed to BEA WebLogic 6.1 J2EE application server, and exposed to diverse global JPMorgan credit trader clients (NY, London, and Hong Kong blotters) as web service SOAP end-points. Spartak SOAP layer would take receipt of global credit traders’ blotter inputs, capture/translate the trade via EJB components into internal business object model, interface with JPMorgan’s internal credit trade pricing, risk assessment, and FO & MO authorization messaging systems (BEA WebLogic JMS – Tibco RV Bridge). Spartak also responsible for transparently persisting approved credit trades from the internal business object model to the global credit derivatives prime record Sybase RDBMS (system of record). Designed XSLT adaptors/filters to transform internal JPMorgan credit derivatives XSD into ISDA-compliant FpML 4.1 (and vice-versa).

**Global Debt Servicing Division, Ecommerce Group** Designed and developed secure J2EE intranet applications to audit and administer Call Center Support Staff access to client account information within JPMorganChase’s USD$360 billion Debt Servicing business function. Designed and developed J2EE public internet application that provided multi-scenario function Debt Amortization calculator service (schedule options, fixed payment scenario, term adjustments). Amortization Calculator scheduled to be deployed into JPMorganChase public internet production service (chaseonline.chase.com) early summer 2003. Assisted in design of transactional “JPMC FastPay” service, implemented as distributed JMS and EJB2.0 Message Driven Bean components, brokering XML message payloads, that interface with firm’s backend legacy IBM MQ systems of record. Prototype successfully deployed to an internal 3-node IBM WebSphere 4 cluster. “JPMC FastPay” empowers retail and institutional clients to securely perform real-time transactional operations (payments, schedule modifications, support/audit functions, administrative updates) to their accounts via any web browser over the public internet.

**Environment**: IBM WebSphere 4, EJB 2.0, JMS, J2EE Container Managed XA Transactions/JTA, JDBC (Oracle 8.x, jDriver XA), JSP/HTTP Servlets/Jakarta Struts (MVC model 2), XML (DOM/Apache Xerces), JAAS, Ant, WSAD Eclipse 2.1 IDE.

***IT Consultant, Bear Stearns & Co., New York, NY (Jun ‘96 – Oct ‘02)***

**Capital Markets IT**

Designed and developed distributed object platform infrastructure and GUI applets for delivering enterprise legacy application data stores to end users' web browser desktops. Consulted numerous, specific, Bear Stearns technology teams (Financial Analytics, OTC Derivatives, Credit Risk, Equities, WRAPS, and Reuters/Telekurs/Thomson/Muller – Global Equities Price Feeds) on best practices and design patterns for implementing this distributed architecture in their core business applications using the BEA WebLogic Java application server. Project showcased the very latest Java 1.1.x distributed computing technologies including HTTP servlets, EJB, RMI, JDBC connection pooling, and Java JFC, Swing. Project also included the design and development of a highly secured, single-login, authentication scheme whereby client-side digitally signed X.509 certificates are processed by a middle-tier Java VM servlet to establish client identity, ACL empowerment, and the cryptographic parameters needed to communicate securely via SSL 3.0 protocol..

**Environment**: Solaris 2.x, Windows NT 4.0, JDK 1.1.x, Sybase XI SQL Server RDBMS, Rational Rose/UML, Sun Java Workshop IDE, BEA WebLogic Tengah 3.x Java application server, Sybase jConnect (type 4) JDBC drivers.

***Application Programmer and Consulting Architect,* Bell Laboratories, AT&T, Verizon**

***(Jun ‘85 – Jun ‘96)***

Numerous, proprietary, telecommunications software development projects.

**Environment**: ON-line Provisioning, TASS, 5ESS, CPSC-Intelligent B/S/K Circuit Management, TOPCOM, SOAC, CRISS, PAWS, OTS/Epic, Unix System V/Solaris, C/C++ programming languages, TCP/IP, Korn Shell, awk, et. al, Unix scripting tools, X-11 Windows/Motif, Tcl/Tk, Sybase RDBMS, BEA Tuxedo Transaction Processing Monitor.

**Qualifications**

**Rutgers University, New Brunswick, NJ**

Bachelor of Science – Computer Science (graduated high honors with distinction)

**Affiliations**

Mar ’11 – Present **Java Community Process** JSR-107 Expert Group (Java Caching API)

Feb ’03 – Feb ’05 **Java Community Process** JSR-156 Expert Group (JAX\_WS Transactions API)

Jun ’07 – Present **JBoss Community Developer** JBoss XTS Code Committer