**Hands-On Enterprise J2EE Architect. Canadian and EU citizen. SC Clearance.**

Enterprise Architecture for large projects in Java/J2EE/EJB/XML/JMS or C#, C++, .NET, SOA; Weblogic; Oracle

Coherence, GemFire, Cassandra; JRules/Aion//PROLOG/Scala/Spark; Rule-based systems; RUP/TOGAF/ZACHMAN

***Recognitions*** COMDEX Best Application in the World Trophy, with the Merrill Lynch team;

RON technology candidate of the year list, with over 500 member companies;

Individual Liberty Mutual Award for critical project contributions, with IBM;…………………...

***Skills*** Distributed Multi-tier Architectures; SOA, CORBA/COM/DCOM; C++; C#, JAVA; Rule-Based

Systems, ILOG, JRules, JViews, PROLOG, Smaltalk; Grids – Coherence and Gemfire;

Scala, Spark, Akka, Play Fwk; JMS/JTS, XML; WebLogic, JMS, UML – RSA, TogetherJ.

***Certifications*** Functional Programming in Scala – Odersky/Coursera (92% with distinction) Oct. 2014

C# - .NET Proficiency Certification by TeckChek May 2009

Project Management Certification by Brainbench, Aug. 2006

Enterprise J2EE Architect Authorization by Derrico Computers, Jan. 2004

EJB/J2EE Certification by Brainbench, Cert. ID3289262 Oct. 2001

SUN Java 2 Platform Certification, Cert. FA4DTT19C2 Dec. 2000

***Author of*** “The 8 approaches for Coherence Caches Loading and how to choose the right one” - 2011

“Visitor, Observer, and Chain-of-Responsibilities Patterns in Oracle Coherence Grid” - 2009

“The 30 Performance Checks for Oracle Coherence Cache Data Grid” - 2009

“The 25 Steps for Developing an ILOG JRules Enterprise Web Service” - 2008

“The 18 Enterprise Architecture deliverables” - 2007

“The 60 Best Practices of Development with Weblogic Integration (WLI)” - 2005

“ILOG JRules vs. PROLOG – forward with Rete or backward with Backtracking” - 2005

“The 85 Oracle Performance Bottleneck Checks”, - 2004

“The 28 RUP Templates for Medium Projects – reducing the 43 RUP Templates” - 2003

***Training*** Toronto, CA Functional Programming in Java 8 - 2014

Toronto, CA Functional Programming in Scala (Odersky-Coursera) - 2014

Toronto, CA Big Data Map/Reduce with Hadoop /HBase - 2012

Detroit, MI Oracle Fusion 12.0 development with WLS 12 and Coherence 3.7 - 2011

Scottsdale, AZ Oracle Coherence (Tangosol) Performance Optimizations - 2009

Toronto, CA Oracle UCM (Stellent) Content Management Best Practices - 2008

Singapore, SG JSF/Ajax development for Weblogic Portal 9.2 (Ajax4Jsf) - 2006

Singapore, SG Integrating Weblogic Components with Aqualogic ESB - 2007

New York, NY Clearcase Best Practices for Weblogic 8.1 projects - 2004

Toronto, CA J2EE 1.4 Application Development with WebLogic 8.1 - 2003

Toronto, CA Developing Microsoft Visual Studio .NET (C#, XML, SOAP) apps - 2002

***Education*** Ph.D. Thesis "CONTEXT AI Expert Systems Development System" built in Prolog

Thesis published in Japan, Germany, France, Greece, and Hungary by IEEE, ACM.

MSc. Mathematical Modeling in Computer Science, from Sofia University .

Bachelor in Mathematics, from Sofia University .

**FINANCIALS**  Barclays Capital, Scottrade, Bonddesk Group, Merrill Lynch, Fidelity Investments, Bear Stearns,CIBC

Wood Gundy, Toronto Stock Exchange, Capital One, Liberty Mutual, CIBC Trust, Aetna, Option One.

**TELECOMS** Apple Computer, British Telecom, Vodafone, Nokia, MTN Telecom, AT&T, Cingular Wireless, Telus

**FORTUNE500** Oracle, Barclays Capital, KOHLs, Lloyds, Kuoni, Shell, Apple, BEA Systems, IBM Global Solutions, IBM Corporation and Labs, Nokia, Vodafone, Liberty Mutual, Accenture, Intuit, Toronto Stock Exchange, Telus Communications and Edmonton Tel, CIBC Trust, CIBC Investments - Wood Gundy, Liberty Mutual, Fidelity Innvest.

**Projects History**

~~~~ FOR: **IronShield ~**~~~~~~~~~~~~~~~~~~ INDUSTRY:  **Security / Cryptography ~~~~~~~~~~~~~~**

Oxford, UK Tools: Scala, Akka, Play Framework, Spring/ XD, Web Sockets, Encryption

Dec’14 – March’15 Project: " Scala/Play Implementation of Conference Security Protection"

Implementation engagement to code a Scala version of a Full-proof-protected Conference Room Application in Play Framework (MVC over AKKA) using the company's Secure Protocol Communication layer over Web Sockets. The Encoding algorithm makes the 40-year-old Public-Private Asymmetric Cryptography as obsolete as it should be, with all the inherent performance problems. Security made theoretically unbreakable - NSA would be just as helpless to decode the dataflow as helpless the designers and me (the implementer) are. The Application class renders both the encoded message AND the actual decoded message, activated via JS through the routes control-file, the JS requests being handled by the Web Sockets interface using Scala Lists. Implementation conducted using Scala-mapping to parallelize the processing, and Scala-foldLeft / mkString to assemble the messages in the onMessage() method. Broadcasting via the standard notifyAll(). Multiple messages merge in the same encoded dataflow, as well as the same message splits in different dataflows, thus deflecting attempts for brute-force key-search. Made the implementation embeddable in layered encryption as a component of a larger standard encryption framework - experimented with Spring XD streams (Spark).

~~~~ FOR: **KOHLs** ~~~~~~~~~~~~~~~~~ INDUSTRY:  **Retail ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~**

Santa Clara, CA Tools: Coherence 3.7, Scala, Spark, ATG commerce, OEM, Spring, Play Fmwk

Jul’14 – Nov.14 Project: "Coherence Re-design and Performance tuning of KOHLs Commerce"

Re-architecting company's RESTfull OpenAPI from EHCache into Coherence. Sizing the Static caches (Product, Inventory, Price, Promotion, etc.). Coherence-type Push-replication POC on 2 EHCashe-DCs through a hole in the Firewall via TCP-sockets on the hole-port, eliminating the Buffers causing the trouble in the Coherence Push-Replication operation. This also eliminated the time-discrepancy in the replication via a Buffer since all put operations are written on both ports - the Cluster-definition-port and the Firewall-hole port. Just for testing purposes, configured the Cluster-definition-port to be the same in both Data Centers, both going through the Firewall hole and forming a WAN cluster (impossible in Coherence). Designed the Dynamic Data Caches (related to Customer / Basket / Order / Billing / Shipping caches) and their cooperation via utility-cache listeners, off-loading the listeners to work-threads to shield them from becoming a bottleneck. Designed 4 business-specified POCs in the new Coherence Environment to measure the different types of load-related performance with Scala Array data-generators with parallel map-functions to adjust the business-dictated value boundaries. Created a Spark Scala Shell poc over a Customer RDD with a Filter-transformation, with a reduce and collect actions on the resulting RDD (count). Designed a Relevancy Cache keyed by ProductID and providing direct access to a Set of relevant products to be fed into a business-rule engine for context-filtering. Experimented with reduced number of Server Nodes to test and measure the influence of the Network chatter on the performance. Adjusted the socket buffers, high-units, JVM parameters to fit the memory requirements and the network capacity. Compared Coherence vs. Spark operations and integrated Coherence with Spark in order to make Spark fault-tolerant and avoid the Backing Map Coherence performance bottleneck, with 2-way-converters Java – RDD – Java to serve the Spark – Coherence – Spark data flow. Implemented a Spark version of “writeBehind” where the bulk operation goes from Spark to Coherence rather than to a disk. Web tier (MVC) alternative approach to ATG via Play over AKKA actors, to match the Scala alternative platform.

~~~~ FOR: **Kuoni Global Travel ~**~~~~~~~~~~~~~~~~~~ INDUSTRY:  **Travel / Retail ~~~~~~~~~~~~~~~**

London, UK Tools: Coherence 3.7, ATG Commerce, Endeca, Scala, Infinispan, Spring.

Jan’14 – May’14 Project: "Coherence Re-design of Kuoni World-wide Reservation System"

Complete re-design and implementation of the Hotel Reservation and Management system away from JBoss/Infinispan into Oracle Coherence. Designed and prototyped the caches needed to hold the data for the entire reservation system. Designed and implemented asynchronous solution for a single Gateway architecture with input and output caches operated via Cache Listeners and a single-service co-located caches cooperating via backing map queries in the Pricing Engine for the RatePlan / RateRule caches, de-normalizing the Property – Contracts relationships in the process, allowing for proper operation of the extensions for the 3 different Pricing Engines (Property, DI, DS). Outside data updater application for reflecting data changes in the database from external applications, eliminating the need for Golden Gate adapter. Designed and implemented a synchronous solution for multiple Gateways, wrapped in a Java Service, distributing the Searches after the ATG Commerce layer between the Endeca Engine in front of the Coherence Layer and the Coherence apps. Prototyped different cache loaders in Scala for convenient data-generation during the prototyping stages of the project. Designed an XML-fragments cache needed for high-performance handling of the assembly of XML pieces from different providers in a fast string-based way. Eliminated a major bottleneck in the operations via re-designing the event- handlers during massive data re-loading.

~~~~ FOR: **Barclays Capital ~~~~**~~~~~~~~~~~~~~~~~~ INDUSTRY:  **Financial ~~~~~~~~~~~~~~~~~**

New York, NY Tools: Coherence 3.7, Coherence\*Extend, Google Guava, Guice, Spring Framework

Sept – Dec’13 Project: "Barclays Capital Risk Management System Extensions"

Risk Management augmentations and business/performance improvements on Barclays' Risk Management System - one of the biggest of the 15 Coherence Projects I have been on with over 200 packages and 3000 classes. Created a Monitoring Service on the Valuation Results cache via a Utility Cache, whose dynamic contents controls the process. The monitor releases another Batch of Valuation Runs to the engine while it is still processing the end of the previous submissions (five different implementations via MapListener / notification, Thread join, poling through configurable interval, Thread ExecutorService/futures, and MapTrigger). Reduced the JMX contention by extending the MBean-Filters with specific types of Caches, lowering the refresh-timeout and increasing the refresh expiry to reduce the contention in the locks on the common mutex in the MBeanConnector. Increased the tenured space of the JVMs to accommodate larger JMX buffers. Created a Polymorphic Cache for storing hierarchies of objects (no direct support in Coherence for that) with trigger-filtering and dynamic down casting. Designed and implemented a Dynamic-Inheritance cache for timely destroying the caches created on-the-fly for handling the Reference and Valuation Contexts, thus easing the space-contention on the tenured heap. Created a set of Entry-processors and nested filters to use in the process, using Google's GUICE Module/Config and Java Dependency Injection. Created a set of self-cleaning and self-destroying Caches with expiration-enforcer for the dynamically-created Reference and Valuation Context Caches - both Intraday and EOD - using Google's Guava Collections for extended functionality. Wrapped the collections to avoid the Serialization Problem while passing through Coherence \* Extend proxies and used readCollection / Codec for template, typed POF read. Created a dual implementation reading the stats in the database via the Spring JDBC Framework and to enable switching between Coherence Caches and the Monitoring Database.

~~~~ FOR: **Scottrade/Oracle ~~ ~~**~~~~~~~~~~~~~~~~~~ INDUSTRY:  **Financial ~~~~~~~~~~~~**

St. Louis, MO Tools: Coherence 3.7, Coherence\*Extend, Coherence\*Web, OEM, NIO

Sept'13 Project: "Coherence Extend / Coherence Web Trading Modules"

Back to Oracle with another Financial Client (Scottrade) for a short Coherence Architecture review. Reviewed the two Coherence Modules of the Online Trading Application including Client, Proxy, and Cache Servers configurations and scripts, monitoring facilities and alternatives, optimized the thread-pool sizes on both the Proxies and the Cache Nodes, corrected the memory assignment on the Proxies, adjusted both the connection and request time-outs, enabled outside modifications through the System Properties, redesigned the topology of the horizontal and vertical scalabity and relocated the Proxies on a single machine, reduced the number of Server Nodes and increased the number of Proxies while reducing the thread pool sizes to reduce the CPU contention and context-switches, provided alternatives for NIO extension of the JVM memory. Arranged the main parameters to monitor in the grid Control in 7 groups and planed the transition to the Grid Control 12c with cloud and SQL-server support, etc.

~~~~ FOR: **CF Retail Chain ~~**~~~~~~~~~~~~~~~~~~ INDUSTRY:  **Retail Chain ~~~~~~~~~~~~**

Atlanta, GA Tools: Coherence 3.7, Spring, REST, Amazon Cloud, OpenID, API-Gateway

June'13 - Sept.'13 Project: "Universal Identity Management in Amazon Cloud"

Architected and Designed the new Identity Management system in Cloud environment (Amazon EC2 and AWS) with the Amazon Cache (ElastiCache). Created a POC with Elastic UserProfile and ApplicationAccess set of dynamically-extendable cooperating Caches, first in Coherence, then in Memcached (ElastiCache) - wrapped in a Proxy-service with custom API-Gateway in front to filter access from both Apache Tomcat and internal components on the App tier. Designed active Chained Cache Loaders via Triggers, and Distributed Authorization via the extendable Elastic ApplicationAccess Caches (each authorization fluctuating between Application-centric and Centralized). Private Session-Management Cache for the Web Tier (Tomcat) for multi-server utilization, accessible from different tiers. Federated ID on SignOn (Yahoo, Google, FB, Twitter, LinkedIn) using OpenID and OAuth. Authentication and authorization via AccessIQ, SAML, and UnboundID, communicating via REST API calls. Implementation in Spring Tool suite with ROO console, with ESAPI, MVC, Mobile, and Spring Data add-ons through ROO. Deployment via the Amazon Elastic Beanstalk with CloudWatch monitoring. Security enhancements against Injection, Session manipulations, XSS, Direct references, and Internet bots. POCs with alternative NoSQL environments (Membase and Cassandra).

~~~~ FOR: **Capital One ~~**~~~~~~~~~~~~~~~~~~~~~~ INDUSTRY:  **Financial ~~~~~~~~~~~~~~~~**

Richmond, VA Tools: Coherence 3.7, WebLogic Portal, Oracle, Spring, P13N, REST, NIO

Feb.'13 - May'13 Project: "Performance Tuning of Universal Portal Framework System"

Reviewed the WebLogic Portal P13N configurations with Coherence as Cache Provider - with Spring Portlet MVC and Spring-aware Coherence CacheFactory, along with Spring HTTP Invoker (for Objects) and Spring Burlap (for XML docs). Prepared the Capacity-planning and Topology-changes Table for the Coherence Resources needed as the load increases. Evaluated the Remote-Node alternatives - Extend vs. Cluster Push Replication. Prototyped the Push-replication leak-fix with Double-push replication. Re-organized the Near Cache memory to use Overflow Map with NIO memory extension to ease the GC contention. Switched to Paged NIO to optimize the NIO Eviction. Pro-con considerations of using Coherence\*Web for session data instead of having "manual" consumer-producer caches with delayed session-expiration. Optimized all 4 P13N Config files for smooth access to Coherence through the Weblogic Portal with Coherence as Cache Implementation. Created a farm of independent Coherence Clusters over the Paged NIO facilities for operating in silo. Cleaned-up the binary unit calculators for both BDB and NIO memory. Re-focused the JMX-based Monitoring facilities to restrict the amount of jmx-related load while keeping the key indicators active. Optimization of the explicit Locking via Entry Processors for Status updates. Documented the testing findings and the performance improvements on both Session and VCR caches. Prepared custom cross-caches monitoring and flush utility nodes with thresholds-based alerting.

~~~~ FOR: **Grid Fluid ~~**~~~~~~~~~~~~~~~~~~ INDUSTRY:  **Financial Middle-tier Infrastructure ~~~**

Oxford, UK Tools: Coherence 3.7, JRules, Prolog, GemFire, CEP, Spring, Prolog/Scala/Akka

Oct.'12 - Jan. '13 Project: "Performance Rule-based High-frequency Grid Trading System"

For a financial spin-off company, designed an Spring-based Coherence Grid integration with 2 rule-based systems: Forward-chained (ILOG JRules) for the algorithmic rule-based trading and Backward-chained (Prolog implemented in Scala/Akka) for Matching Engine (see the Bond Desk Fixed-Income project in 2011-2012 where we used portions of this Coherence Integration technique to achieve massive performance improvements). Utilized the internal synchronized IlrConext thread-pool for double-distribution (Nodes plus Threads) while using a Private Thread-pool for the consolidated Matching Engine - both integrations done via Entry Processors. JRules consumes only references to the objects in the Coherence Caches. All Caches are loaded by the Backward-chained Matching Engine, hooked to the built-in Unification Mechanism. Dynamic JRules Rulesets re-loading, based on pre-specified alerts, with dynamic transfer of Rules from one IlrConext to another. An unique project providing a smooth division of labor between 2 different types of Rule Engines, indirectly cooperating through a set of Coherence Caches to provide a robust CEP event processing. Attempted and ruled-out a GemFire re-implementation due to the "single-cache in single-JVM" architecture of Gemfire.

~~~~ FOR: **Apple** **Comp**. **~~**~~~~~~~~~~~~~~~~~~ INDUSTRY:  **Computer / Telco** **~~~~~~~~~~~~~**~~~~

Cupertino, CA Tools: Coherence 3.7, Push Replication, Golden Gate, DB-reflection, GemFire, Oracle

March'12 - Oct.'12 Projects: "Performance Adjustments", "Apple Fault-Tolerant Coherence Environment"

**Multiple Coherence projects** (see below).

New project: Critical Performance Improvements in several Coherence projects within Apple: OBS (Order), PMT (Payments), CAS (Customer Service). Changing and expanding the JVM and Network parameters to make the cluster course-grained thus reducing the internal communication in the WKA-list-based cluster. Reduced the amount of Entry Locking causing contention. Reduced the number of nodes per VM (VMware) to avoid the swapping performed by Linux, which in turn was causing Nodes Leaving. Changed the Configuration and Startup scripts to adjust the Units, to optimize the number of Partitions, and the Caches topology. Created the Coherence Interaction Models for all the 3 applications, leading to Coherence Processing modifications. Suggested re-implementation of the code requiring Locks via Entry Processors. Prepared a comparison between Coherence and GemFire as alternatives tools for future projects and a selected list of benefits for moving to Coherence 3.7, with emphasis on the Elastic Data JVM memory expansion. Re-implemented the main Coherence constructs used in the Apple's Coherence Projects in GemFire, including triggers (via CacheWriter), Unit-testing Components (via registerInterest() of CacheListenetrs), Continuous Queries (via newCq() ), Custom Queries (via OQL), WriteBehind (via GatewayEventListener), Entry Processors (via Function Execution Service), keyAssociator co-location (via partition-attributes XML), utility tag-caches (via extended keys), debugging listeners (via registerInterest), etc. Conducted a review to assist with the decision for transferring the operation systems in VMware environment. Comparing Coherence and GemFire to open-source Hadoop/HBase, Apache Cassandra, Terracotta BigMemory, and Membase /Couchbase as alternative technologies.

New project: back to Apple for an initial Design and Prototyping of a new Fault-Tolerant Grid Environment initiative. Prepared the initial Design for the Coherence-based POC using a sample Oracle DB with some 100K records. Generated the Hibernate Java Classes and prepared the CacheStore methods for loading the Cache of aggregated POJOs. Designed and prototyped a Client component implementing UpdateBehind (similar to WriteBehind) for reflecting outside Updates in the DB, using the same delay specification to eliminate the congested Trigger-based solution currently in place in Apple. Designed and implemented Double-Push-Replication to provide full refill after one of the Clusters goes down completely, including the Entries pushed through the other side before the moment the Cluster went down, with the two Replication Sites sitting on different computers. Matching the Golden Gate bi-directional replication in Coherence without the need for Adapter. Deployed the POC on 2 clusters, 2 machines each with 12 vertical JVMs. Included versioning in the project (via Evolvable interface) enabling switching back to stable versions. Prepared the XML configuration files for selective JMX monitoring.

~~~~ FOR: **Bonddesk ~~**~~~~~~~~~~~~~~~~~~ INDUSTRY: **Financial ~~~~~~~~~~~~~~~~~~~**~~~~

Detroit, MI Tools: Coherence 3.7, Spring, CEP, Bond Engines, Fusion, Prolog/Scala/Akka, C++

Oct'11-March'12 Project: Grid-based Bond-trading Evaluation Engine 2.0

Reviewed the C++ Engine code and the Oracle Coherence Bond Pricing Services and Caches through the usual 90-point-checklist of config files, network settings, app-logic, and coherence Java code. Prepared the Object Model (in EA - Enterprise Architect) of the Coherence Classes (over 100 of those) and suggested changes in the structures. Designed and prototyped a parallel Matching of packets of Bond Offers against a Cache of Watched Queries, integrating a Prolog engine written in Scala/Akka. Embedded an Unification Engine into the Coherence Server and implemented distributed matching as an Entry Processor, running against a dynamic set of Filters over a Reflection Cache (based on Offer parameters) to substantially reduce the amount of hits. Results put is an Optimistic Cache on a separate Service to put EPs on a separate Service Thread. Bundled the incoming events (Bond Offers) in packets of 10 before setting the Entry Processors, and pre-processed the Watched Queries presentation in the Cache to further increase the performance within the 30-server-nodes cluster. Increased the performance of finding the matching Offers and Watched Queries 1,000 times (3 orders of magnitude) compared to the previously-used brute-force single-process single-thread approach of Bond-Offer matching. Prototyped an Event Translator converting the incoming Offers into simplified internal structures held in a separate Utility Cache, handling the incoming events in Event Stream Processing (ESP) manner with a Reverse-DB approach (storing the queries in the Cache, rather than the records) to provide a CEP event processing from multiple Coherence caches (offers and queries). Re-implemented the Matching via embedding the Esper Java Matching Engine into Coherence, hitting it with another Entry Processor distributed towards a potential set of 1 MIL watched queries per Cache, over the same 30 servers. Sample-data generation written in Scala for the test cases.

~~~~ FOR: **Oracle Corp. ~~**~~~~~~~~~~~~~~~~~~ INDUSTRY: **Financial /Telco ~~~~~~~~~~~**~~~~

See below Tools: Coherence 3.6, Spring Framework, Weblogic, UCM, IDOC, Oracle June’10 - Oct.'11 Fusion, Evident, Wily, JRules, JViews, Spring Batch, Multithreading

Back to Oracle as hands-on Architecture Review consultant in Oracle Coherence and Oracle Fusion.

- For a **large Computer Manufacturer in Cupertino, CA -** Conducted the initial code-review and configuration changes for the "Apple Online Store - 2.0". Designed and implemented a series of Unit Testing Modules with utility Triggers and Listeners to "cut" the process in pieces for proper independent testing. Resolved the dead-lock issue by resizing the pools and cache sizes, changing the service.config, and introducing an utility cache to store intermediate results. POC for Delete-behind with 4 different approaches. Changed the put/get blocks via intermediate HashMap and nulled the return values for the Entry Processors where values is not used. Designed an (n+1) counters system to release the implicit locking contention by the Entry Processors, and a Custom Eviction policy collecting the evicted objects, as well as a Cross-listener for filtering Synthetic events, related to the Eviction problem. Designed a MapTrigger approach to getting NEW Orders in Apple during DB operation without blocking Orders via a Buffer Cache. Resolved the memory problems resulting from huge results sets via custom Aggregators and paged queries. Prepared a detailed Interaction-Diagram for all Domain and Utility Coherence Caches, with all Listeners, EntryProcessors, and WriteBehind events involved. POCed a Reporting Tool via both Invocation Service and JMX monitor walking the entire JMX tree. Prepared the entire set of Unit Tests for end-to-end unit testing, as well as Inside Unit Testing via series of Unit Listeners, Triggers, and Aggregators, along with the additional static data needed to the Inside Unit Testing. Prepared the Class Diagram for the Tests with the Starters, Listeners and Entry Processors.

- For a **large Oil Company** **in Calgary, AB** - Coherence Architecture and Implementation of the "Enbridge Pipeline Scheduling System". Co-created the requirements-driven Models in TOGAF-based mode of EA (Enterprise Architect). Designed and implemented the Pipeline Building Block with a quintet of Coherence Caches (four Activities Caches and a Floating Cache holding Fluid Objects which sit partially in 2 caches), from which any Pipeline can be constructed. Designed and implemented a Cross-Caches Coherence Entry Processors executing synchronously across a Chain of Caches to advance Fluid Batch Objects across the Pipeline. Designed and implemented Revolving Coherence Caches for Pipeline Scheduling Tune-up and Routed-Allocation-based seeding of the Rule Engine (ILOG), using JViews Schedules for the scheduling presentation console. Used the Reversed MVC pattern (multi-models with single view) to implement Multi-scheduler Terminals editing the Master Schedule in a private copy, later pushed into the Schedules Cache for processing, optimization, simulation, metrics, and approval for promotion into a new Master Schedule, with private Work-Schedule Caches pointed to by outside Schedule Viewers. Designed the Schedules Cache as a Merging Coherence Cache for consolidations of multiple editions of Candidate Schedules. Developed a PROLOG-based Scheduling prototype in JIProlog for simulation before the processing by the ILOG CP Engine (a PROLOG-based engine by itself). Developed the XSD / XML for Pipeline Definition and Configuration (in Stylus) and for intra-terminal Facilities Definition (tanks, segments, valves etc.). Serialization automation using XStream as a WriteBehind mechanism while Loading via TopLink, and IOC via Spring.

- For a **large Oil Company** **in Houston, TX**, - Coherence Architecture Review and Go-Live activities of the "Shell Global Trading and Risk Management Platform based on ADS" (best Coherence Application of the Year for Oracle) in cooperation with the Oracle RAC team. Reviewed the Network configurations, Cluster topology, Proxies and JMX servers, start-up parameters, etc. through a 55-point Performance Optimization tune-up. Alternative Aggregation approaches via a Cross-Caches Collector Cache and Custom Aggregators (via ADS layer on top of Coherence). Suggested optimal vertical scalability structure of the Nodes and optimal memory capacity, as well as changes of the existing topology of the Cluster. Prepared a project Coherence-Check-List and Action-Plan in case of future potential problems when the Oracle team is no longer on the project.

- For a **large** **Financial Institution in Charlotte**, NC, US - Lead Coherence Architect for their fundamental re-architecture of the entire IT infrastructure, changing it to Coherence and Weblogic (Oracle Fusion, Portal, Coherence 3.4 migrated to 3.6.1). POC of 7 GOF design patterns in Coherence, evaluation of Evident, RTView, and Wily Introscope as potential candidates for Monitoring tools. POC covering over 20 specific features of Coherence as potential building blocks for the future re-engineering of the enterprise. Reviewed the Coherence Topology in the Bank, the Main Caches and their utilization, and presented the alternatives in loading the Caches. Presented the Avenues for injecting Coherence into ODS in the Bank. Redesigned the ILOG JRules Integration via Coherence. Design and Develop a Working Prototype on the JRules Cache with Filters and Custom Queries. POC using the CacheLoader Design Pattern of Coherence and design of the automatic Pre-loading of the Reference Data Cache. Design for 3 different Cache Loaders (Portfolios, Funds, Stocks, Quotes). POC for a Cache Loader with Parallel Distributed Processing and Multithreaded Invocable. Prepared a full-blown Coherence Training Facility for the extended team. Developed a POC project with the 16 main Coherence Features for the Training Facility (working prototypes). Optimized the Wily Introscope metrics for Coherence for MIN CPU overhead via the new JMX enable/disable switches. Spring IOC implementation of decoupled Client-Server Coherence processes. Re-designed the Reference Architecture with a new System-level Cache used by the Invocation Service. Re-designed the Reference Architecture - new structure of the Caches Client/Portfolio/ODS/Funds/Contracts with Chained Delayed Loading. Created a configurable Security for the Caches based on UserGroups with Segmented Securitization on Cache Method Level (also configurable) to provide Fine-Grained Cache-authorization. Provided alternative Access-Control via Account PIN-numbers through a system-cache (also used in the Multithreaded Invocable). Prepared a Black-out Cache with zero-access (nobody can GET any Entry) and a Tag-registry cache with a series of Triggered Call-backs performing Cascading Loading of the respective Domain Caches. Designed a Delta-Loader for Changes-only load and a Flip-Pair of Miror-Caches for massive reloading.

- For a **large Telco in London,** UK. Back to Vodafone for an urgent Oracle UCM requirement to finish the new Vodafone public website. Based on the HTML Templates designed and implemented the IDOC script in Site Studio for the Site Navigation (meta, primary, secondary) and the testing site for the Navigation with the approriate test-cases. Designed and implemented a user-profile-based left-side navigation which extracts only the sections corresponding to the current user profile in Weblogic Portal. Prototyped the Portal integration via the wcs\_placeholder service with region templates. Extended the user-based flexibility with IDOC array of personalized menu items and personalized order using reMake from strings. Implemented the IDOC script for all the head-sections (title, meta-tags, css etc.) getting the elements from the idc server. Made the user-driven menu controllable from the project file via a set of Custom section properties. Using Scrum extended with XP Agile development and the new DevOp approach, implementing via small iterations on user-stories and frequent releases.

~~~~ FOR: **Lloyds Global Markets** ~~~~~~~~~~~~~~~~~~~~~ INDUSTRY: **Financial / Investment**~~~~

London, UK Project: ” Caspar Market Data Processing (Coherence hub)”

April’10 – June’10 Tools: Coherence 3.5, Java, C#, Coherence Extend, Oracle DB, RTView

Designed and implemented a Persistent Binary Store Loader from flat files via a configurable thread Pool and the Java6 Concurrent Package to shrink the loading process 10-fold. Wrote a pre-processing XML-manipulation function to perform a variety of CRUD operations on the config files in order to avoid the non-standard Java Sax parser. Triggered on-demand Heap Time-Slices on the nodes via HProf for tuning the cluster. Wrote a custom Eviction Policy class to trace and tag the evicted objects, as well as a Member Listener to trigger a configurable restarting of nodes leaving the cluster for too long. Wrote a Bulk-loader of POF-objects both from files and from a DB Table ( putAll() of linked HashMaps). Wrote another Member Listener to monitor the JVM-memory of the nodes and alert when the yellow-zone is reached. Optimized the network settings to achieve max performance of the datagram-test and the multicast-test in both modes (client and server), as well as the number/size of the partitions of the cluster. Reconfigured the cluster’s vertical and horizontal scalability schema to improve the throughput, moved and reconfigured the Proxy Serves, splitting them in groups to isolate the C#-client usage, both in terms of servers and grouping of the ports-ranges. Isolated the JMX server and wrote an AlocateDirect test to find the NIO boundaries for the Proxies. Redistributed the memory for the caches according the cache sizes and increased the thread-pool sizes as needed for optimal throughput.

~~~~ FOR: **Oracle Corp. ~~**~~~~~~~~~~~~~~~~~~ INDUSTRY: **Financial / Education / Market**~~~~

Various locations below Project: ”Oracle Architecture Review consultant (Coherence / UCM)”

Nov’08 – April’10 Tools: Coherence, Oracle UCM, Oracle Fusion, Jive, SOA,WebLogic Portal

**ORACLE** Architecture Review consultant for various Oracle clients **Oracle Fusion, Coherence, UCM (Stellent)**:

* For a major distribution company, participated in the initial off-site assessment of the current PO-systems, followed by an on-site prototype of reengineering the business in Oracle Fusion (SOA-based). Created the initial Specification of the necessary Interfaces and put together a mock-up Java-proxy of the WSDL stubbed-out interfaces to get a quick feedback from the Purchase Order GUI. Designed the first cut of the BPEL Composites and the common in/out XML for the BPEL processes (single process in every composite). Generated the XSD based on approved sample XML. Used plain Database Adapters for both the eBisuness Suit (EBS) components and the plain db-resources (eliminating the EBS adapter). Explored the possible incorporation of a grid structure (Oracle Coherence) to hold the POs in a cluster across servers via a Java Embedding node in the BPEL process.
* For Government Financial body, architecture review of a content-management (Oracle UCM) in Oracle Coherence environment, with Oracle Fusion front-end (WebLogic Portal, rather than Site Studio). Disaster-recovery plan based on Push Replication pattern through 2 remote Coherence clusters, and explored the reusability of the Visitor and Chain-of-responsibility patterns from the previous Oracle assignment (see the next paragraph). Reviewed different ways of communication between Oracle UCM (Stellent) and the Oracle Fusion Portal (WS, RIDC connector, CIS, CPS Portlets, OpenWCM, the standard JCR, and the VCR adapter) and selected the CPS portlets due to the ready-to-use widgets with the CIS behind. Designed a prototype for a content-item Coherence Continuous Query, hit by AJAX on the Portal (XmlHTTPrequests), achieving significant performance improvement on the Front End. Designed a DSS for Coherence Serialization which suggests the optimal Serialization from 5 generated Object Serializations (Serialize, Externalize, Externalize Light, XMLbean, and POF) depending on the complexity of the serialized object and the implementation overhead.
* Architecture Review, again with Oracle, but this time with Coherence Distributed Grid implementation rather than Stellent (after training in March, see the front page). Architecture Review of the online education for Phoenix University system with terabytes of data flowing monthly and up to 50 message per second in peak periods. The system is based on Coherence cache, OS Cache, and Resin – the so called Jive Discussion Forums COTS product. A severe deadlock situation due to running all the above 3 systems in a single JVM with auto-restart based on memory-thresholds. Analyzed the thread-dumps, the sdterr logs, the coherence logs etc. both with TDA and Samurai. Designed the extended Coherence Logging in the Pre-shutdown routine of Resin to collect detailed information for both Front and Back cache of the Near Cache, and then implement a graceful Coherence shutdown before the Resin restart. Prepared a 30-point list of Coherence Performance Bottleneck Checks and a series of test based on smothering the TcpRing heartbeat communication. Optimized the topology of the Coherence network based on the CPU cores and the granularity of cache for Vertical scalability. Functional comparison between Oracle Management Console and the ClearStone from Evident Software – selecting the Monitoring tool for RT-reports. Upgrade the version to redesign and take advantage of Collocation of related entries (Affinity), faster POF serialization and POF Extractor updates with no deserialization, as well as custom-eviction and auto-invalidation synch with the Back Caches. Spring-based assignment of BM Listeners via a set of Inversion-of-control XML files. Used the Coherence Command Pattern to implement Chain of Responsibilities (multiple Command Objects hitting single Context), Visitor Pattern (single Command Object hitting multiple Context Objects), as well as a Command Pattern returning values from the “execute” method (distributed version of the Functor pattern via external HashMap). Reimplemented the Visitor and the Chain-of-responsibilities in a Synchronous way via EntryProcessors. Used the Command as a Façade for dynamic 3-way communication with a JRules Engine component (via JMS, Web Service, and plain Coherence Object), controlled by an incoming parameter reflecting the current system load.
* Architecture Review of the reengineering project for handling Salesforce Communications from VB / MS Access prototype to Weblogic WLI / Oracle / Stellent SOA environment (similar to the Partnership Communication Management in the MTN Project in Cape Town). Stress-tested the WLI Process-in-Process nodes which fork sub-processes for Batch Load of new salesforce documents in Stellent to set the optimal number of forked processes under specific threshold load boundaries. Provided a Stellent Bulk Loader generator for both Oracle Data (via JDBC) as feeds for the JPDs and straight directory structure with automatic Metadata extraction for the cron-jobs. Revised the entire Site Hierarchy and the underlying Sections, as well as the presentation HCSPs and Fragment Libraries. Reviewed the IDOC Scripts on the presentation layer (Stellent Site Studio) and their reusability in the HCST and HCSF dynamic Server Pages. Factored-out some of the scripts and moved some of the IDOCs to specific event-handlers. Changed some of the steps in the contributors workflow, the ratio of the primary / secondary pages, and the validation scripts for the Contributors in the contributors IDOC. Checked the WLI JPD-files against “The SOA 60 Best Practices” for points of improvement and the SOA communications via the Aqualogic ESB with the services layer (System and Stellent components). Checked the Oracle Settings and Procedures against “The 85 Oracle Performance Bottleneck Checks” for performance optimization of the Back End.

~~~~ FOR: **Ministry TCU ~**~~~~~~~~~~~~~~~~~~~~~~~ INDUSTRY: **Education, Financial** ~~~~

Toronto, Canada Project: ”Loan Management System”

Sept.’07 – Oct. ‘08 Tools: WebSphere 6, RSA 7.0, SOA, WSRP, WS Portal, JSF, Stellent

Lead Architect for a large Government Loan Management system with over 8,000 requirements in RequisitPRO, over 2,000 JRules business rules, cooperating with 300 business partners. Prepared the Detailed Operation Plan for the entire WebSphere development, extending Zackman/Togaf Framework for all the 78 Nodes. Designed 3 integration alternatives between ILOG JRules and BPEL (Web Services, Session Façade, and POJO façade) and implemented prototypes. Prepared two JRules Component testing Java programs, one of them via a wrapper of “executeRules()” call to fit the BPEL nodes. Proxy-filters for pre-processing of the ILOG JRules XML parameters with configurable components (via Spring IC). JRules design according to the Agile ABRD/ISIS Rule standard of ILOG. Evaluated PEGA systems BPM/Rules combination to substitute the alternative of the BPEL-JRules combo. Designed and developed a JDBC-based PROLOG code-generation for the entire AS400 database metadata to run against the PROLOG Schema-assessment AI-facilities – with Java and DDL generation. Reversed engineered the Java code in RSA to facilitate easy Data Migration process. Prepared a Oracle-TCA based sample schema to be used by the assessment programs. Designed a flexible Loan Rollover facility, based on XML-config file. Designed the centralized Global Corporate Security Hub to be used by all satellite portals and implemented a POC for the Interfaces. Transferred the Visual Architect BPEL Models in RSA 7; build a pre-processor for RSA to allow reverse-engineering of JSP/JSF/HTML for Screen-Flow generation. Lead the development of the main POCs of both the Middle Tier and the Portal skeleton, all in IBM RSA 7. Developed a Bulk-Loader Generator for Stellent.

~~~~ FOR: **MTN Telecom ~~~**~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ INDUSTRY: **Telecom** ~~~~~

Cape Town, Joburg, SA Project: ” PowerBill Unified Telecom Billing System”

Jan’07 – Sept.’07 Tools: Weblogic 10, Ajax, ADF, JSF, SOA, ESB, WS, WSRP, DHTML

Lead Architect of the new Telecom Billing System in MTN, consolidating all the current resources via an ESB implementation across the MTN internal systems. Customized some of the eTOM Telecommunication Standard Processes and the SID Telecom Object Model (in Rational Architect) in DDD-architecture (Domain-Driven-Design) to fit some the MTN current resources, wrapping other in Facades in the process. Merging Requirements and Use Cases in IBM Rational Prerequisite PRO Composite project.. Stress-tested several ESB approaches (plain Web Services in a Centralized Hub, Open-source Mule, WLI-proprietary Weblogic implementation) via JMeter and identified certain bottlenecks in a couple of the current systems. Prototyped and re-measured with Weblogic 10.0 / WSRP and AquaLogic. Implemented the first JSF Prototype of the Trading Partner Portal (TCP) using Oracle ADF Ajax-aware JSF components and JDeveloper for both Partner-billing and Revenue-sharing using the new SingleView interfaces and the Service-Object-Request-Broker. Oracle-oriented TCP approach to fit the move to Siebel on the CRM-side using the Siebel eService configurations, Siebel WebUI 8.0 and Oracle COREid Single Signon and Oracle SOA Suit. Parallel JRules and Oracle Rules (JESS) prototypes of Rule-Based User Profiling. Cleaned-up the 40+ ESB Interfaces and consolidated them in different segments to facilitate Proxy-based isolation on the Client Side. Designed the Wireless (PDA) client optional billing interface. Designed and implemented Java Product Catalogue via XMLBeans and SAXON-XQuery and via XML DOM .

~~~~ FOR: **Mansion.com ~~~**~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ INDUSTRY: **Gambling** ~~~

Singapore Project: ”J2EE Version 2 Re-architecture in Weblogic 9.2”

Jan’06 – Jan’07 Tools: C++, CORBA,WSL, Ajax, JSF, JRules, SOA, WS, Spring, Hibernate

Lead Architect of the Poker Component and the J2EE re-architecture of the entire Mansion Enterprise system. Transferred the CRM / Customer Care MySQL and MS-SQL databases to Oracle and checked the existing Oracle Functional DB against “The 85 Oracle Performance Bottleneck Checks”, transforming some trigger-based relationships into declarative-integrity constraints. Reverse-engineered the C++ Poker code base and generated JavaDoc-type of documentation to prepare it for the J2EE transfer. Isolated and optimized the SQL Statements (SQL Optimizer for Oracle 10g and TOAD 9.0). Reviewed the existing CORBA Architecture of the Poker Site and designed 3 prototypes for Resource-Contention and Synchronization performance issues of the contention point in the so-called “Poker Lobby” - via CORBA Event Service, via STL Queue, and via Observer Pattern, geared towards elimination of both the resource-locking and synchronized call-waiting in order to ease the performance pressure during the transition to J2EE. Prepared prototypes (cornerstone apps) in the new Weblogic 9.2 environment (both Eclipse-based and Studio-based). Tangosol Coherence prototype using the Cache and Session-replication facilities with alternative caching using the OSCache for the JSPs. Lead the Architecture Prototypes and the Best Practices documents for Weblogic development. Using the DAL and SAL layers of DevCode, lead alternative prototypes for DB and Service isolation layers. Prototyped an out-of-portal portal, based on JSF/Ajax/Spring with native Struts (rather than the one built-in the Weblogic environment) as well as Jsf/Facelets/Struts approach (using the new Exadel Studio 3.6). Researched the possibilities for using Ajax-aware JSF components in the WL Portal 9.2 (via the open-source Ajax4Jsf tags) and the new Weblogic 9.2 Studio. Prepared a Java Robot in Swing with Substance Look-and-feel capturing all activities from all the clients and researched the possibilities for (and prototyped) the Rule-based AI-modules involved (both in Prolog and JRules). Executed sample performance testing of the Java prototypes with JProbe 6.0 and VTune 8.0 to prepare for using the Linux/C++ version of Intel’s VTune for the C++ performance and profiling, pinpointing further pressure-points for releasing the pressure in the current C++ version. Found and fixed the erroneous RNG-code in the generators. Code-reviewed the main 2 applications via the PMD and CheckStyle source-code plug-ins, and the cross-component JLint and Lint4j. Ran the performance analysis with VTune and the JUnit tests. Researched alternative methods for fraud-detection, including screen-scraping and socket-sniffing, and created modules, blocking both fraudulent methods, involving persistent objects in the new EJB 3.0 (KODO) implementation in Weblogic 9.2 in order to test the new persistence mechanisms. Alternative Java Persistence API app over Hibernate. Prototyped Remote Rule-Based control module with an MDB object of JRules 6.0 for Weblogic, sending the results in a dedicated Pub/Sub topic to replace the Observer Design Patter from the old C++ modules.

~~~~ FOR: **SUN Microsystems ~~~**~~~~~~~~~~~~~~~~~~~~~~~~~INDUSTRY: **Financial / Telecom** ~~~

San Jose, CA Project: ”Medicaid/Medicare System – J2EE Redesign”

July’05 – Jan’06 Tools: WebLogic Integration, SOA, JRules, JSF, JViews, Ajax, SUN Portal

**SUN** Architecture Review consultant for various clients of SUN Microsystems, East and West coast:

- Architecture Review for a critical ILOG JRules in WebLogic Integration (WLI) project for a major Mortgage Company. Introduced a new set of more than 40 new business rules in BAL for the next release of the system; checked the ILR translation to match against the XML Test Dataset and pinpointed some data inconsistencies leading to some bad regression-testing results; prepared several utility functions used by the TRL rules; considered alternative approached of using ILOG JRules in WLI (the ILOG Control requiring a new version 5.1 only; a Web Service control, a plain Web Service, and a plain Java Class running in a Session Bean) and prepared a pool of IrlContexts for efficient rule-engine connection establishment.

- Deployment Review for a major Document Management company – a WebLogic Server (WLS) project; tuned up all the JVM starting parameters of Itanium to maximize performance; checked all the Clustering and Load Balancing decisions, introduced a middle-tier Load Balancer; reviewed the performance implication of using the JSession Ids; prepared a plan for the Code Review, Class Review, and HP JMeter utilization for heap analysis, object creation, call-tree, thread-based metrics etc. Prepared a parallel-comparison with JProbe, OptgimizeIt, PMD etc. Prepared a prototype with Safe JDBC operations avoiding the resource-leaking problems the project exhibited during tests. Prepared a Daly routine for the WLS Admin.

- Lead Architect on the re-engineering effort from CORBA (Visibroker) to J2EE (Session EJB and WS) of a leading Legislations Systems company. Analyzed all current Journal Handling and REF Systems for monitoring the House and the Senate meetings, the Bill Introduction sessions etc. Made the Object Model for the current systems and the Use Case diagrams for the top 5 applications. Factored-out the functions from the previous CORBA interfaces and prepared 5 Application Facades to be used to communicate with the Service Locator and then the mid-tier (session EJBs or WS). Prepared a Rule-based prototype to control the process based on Business Rules on top of the SML Schemas, with cross-Elements and cross-Attributes Business Rules. Implemented the Rules in PROLOG invoked from Java and then in plan Java for comparison. Documented the code and artifacts. An XMLHttpRequest-based client (Ajax) and a corresponding XMLEncoder / XMLDecoder-based Swing clients.

- Lead Web Architect on the ACS Medicare project – redesign of the old COBOL / Power Builder system in J2EE / Web environment. Built the Object Models for the Web-tier to Middle Tier communication on the basis of the 350 Use Cases and Scenarios of the main Use Cases. Built the prototypes for JSF – Spring Framework communication, as well as the Ajax in JSF context. Developed the alternatives comparison between JSF in JViews framework vs. JSF in SUN Portal framework, as well as an evaluation of the prospects of porting it into WebLogic and Websphere for the ACS’s future clients. Prepared a comparison between Hibernate and Cocobase to facilitate the decision for an OR-mapping tool. Prepared the Web Tier Architecture Document for the project, based on the proof of concept mini projects mentioned above, with Class Diagrams for both Portal and plain JSF environments.

~~~~ FOR: **Bear Stearns ~~~**~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ INDUSTRY: **Financial** ~~~~~~~~

New York, NY Project: ”Equities Data Metrics and Risk Management System”

April’05 – July ‘05 Tools: WebLogic Integration, SOA, Cocobase, Cache, Sybase, Blaze Rules

Lead Architect on a very high-profile SOA-WLI Equities Risk Management project on Wall Street. Designed and developed highly multithreaded JPDs (Business Processes) for efficient processing of the daily Bloomberg, ADP, etc. feeds (some of these are over 5GB-files) with MFL/XQuery transformations 10-times faster than the BEA XQuery Engine coming with Weblogic; developed a highly-flexible Implied Volatilities curve-fitting and 3D-surface-fitting algorithm (plain Java in a Java Control, configurable via Inversion of Control – Spring Framework; Least-square fitting); a flexible and abstract Empirical Volatility calculators, STD-calculators, moving averages (for both pricing and volume), data-merging facility for fitting the disparate data from multiple sources (over 160 Prime exchanges alone). Designed and implemented Cache-Based JMS-Queues holding Historical data (needed for all the averages, Empirical Volatilities etc.) and their triggered adjustment upon Corporate Actions; fitted the computations into a grid-based framework (a proprietary grid-implementation) for adjustable control via a rule engine (Blaze), Cocobase OR-mapping, and Gemfire (Gemstone) Cache – a very exciting project.