

**Reid Peryam** +1-206-661-8953 [reid@reidperyam.com](mailto:reid@reidperyam.com)

## **An Overview of Projects I Have Developed For Employers and Clients:**

Cap Logistics, Denver, CO: 2013 – 2017

### **Voodoo Sorcerer**

Full-time employee at \$50m revenue, privately-owned logistics company specializing in international and domestic B2B freight. Introduced and drove adoption of Agile/SCRUM as well as application lifecycle management in transitioning company to support sustainable development goals. Engineered adoption of cloud migration by architecting and implementing hybrid cloud solution. Challenges include integration of legacy data to drive hybrid solution both locally and from the cloud and integrating 3<sup>rd</sup>-party products alongside in-house, proprietary solutions.

1. **Sales Application** – Designs, architects, develops business intelligence and rich-UI CRM web application to drive internal sales team and management sales strategies and tactics. Technologies utilized include C#, ASP.NET Web Api, JavaScript, .Net 4.5, SqlServer 2012, MongoDB, KnockOut.js, JQuery, Azure services (Websites, Cloud Services, Service Bus, Active Directory, Traffic Manager) and a bunch of OSS (ie OWIN, NancyFx, Autofac, XUnit, ScriptCS).
2. **Web API** – Designed, architected, develops API supporting company's evolvable development efforts on the cloud. To leverage historical, proprietary data without transitioning databases to the cloud, a hybrid cloud solution was necessitated. Current architecture is "REST-ish" (Richardson Maturity Model level 2) with roadmap for further hypermedia/HATEOAS integration. Also, engineered hybridized system/integration tests within shared code base. Technologies utilized include C#, .Net 4.5, ASP.NET Web Api, SqlServer and a bunch of OSS (Dapper ORM, OWIN, Autofac, XUnit, ScriptCS, Configr, StyleCop).
3. **Quick Quoting Module** – Developed web-application to drive operational revenue by expediting company's ability to provide service quotes to customers over the phone. Leverages legacy pricing data along with third-party APIs (FedEx, UPS, SMC3, TravelPort) to retrieve and integrate airport and carrier logistical data. Technologies utilized include C#, .Net 4.5, Aurelia JavaScript framework, proprietary internal, LOB applications, SqlServer and the aforementioned Web API application, above.
4. **Customer Portal** – Initiative to automate customer-servicing needs into web application encompassing reporting, shipment tracking and creation, quoting, invoicing and document serving. Technologies utilized include the aforementioned Web API application, above, proprietary internal, LOB applications, SQLServer, Auth0, React JS, Node JS.

Open Source Development, Github: 2012 – present

**Easy Mongo** – A simplistic facade to 10Gen's official C# MongoDB driver. Additional added functionality includes asynchronous support and interface-driven operational usage. Designed to simplify integration and abstract noise of the underlying driver using dependency injection and separation of concerns. Technologies utilized include git, C#, .Net 4.5, MongoDB, NUnit, Autofac, Ninject.

RBA Consulting, Englewood, CO: 2012 – 2013  
**Consultant**

Pearson eCollege, Centennial, CO

Onsite at client, a subsidiary of the world's largest education company. Daily responsibility included sustained engineering for back-end LMS products: Learning Studio, Digital Vellum, Open Class. Working within an Agile team of four responsible for 3<sup>rd</sup> party integrations within product suite including BLTI, LTI, and proprietary APIs of 3<sup>rd</sup> party vendors (eg electronic course book provisioning).

1. **eBooks** – Managed sustained engineering of client’s electronic book provisioning framework as well as third party interoperability. Architected MongoDB logging framework by which to monitor production execution. C# .NET, MongoDB, third party API web services.
2. **BLTI** – Sustained engineering of client’s “basic LTI” (Learning Tools Interoperability) framework to facilitate third party tools to launch from their UX. C#, .NET, MVC 3.
3. **Java/Java Script Tech Stack Transition** – Transitioned team’s deliverables from C#, .Net, SQL Server, MVC to company-directed cloud deployment model including AngularJS, Spring, RestExpress and MongoDB hosted on AWS.

SPATIALinfo, Lone Tree, CO: 2010 – 2012  
**Developer**

4. **spatialWEB** – Responsible for new feature implementation as well as defect resolution for company’s GIS mapping web platform. The product incorporates Silverlight controls engineered with MVVM best practices as well as Bing Maps integration.
5. **ADDRESSmanager** – Responsible for delivering refreshed implementation of geospatially coordinated address merging product. Implemented, documented and tested through multiple iterative versions. Travelled to Melbourne, Australia during six week delivery process. Technologies leveraged include C#, .NET 4.0, Windows services, Oracle database, Oracle Spatial database operations and consumption of PSMA’s GNAF address distribution.
6. **QA Tool** – Responsible for implementing automated, extensible, scalable quality assurance tool to verify solvency and compatibility of new customer data with company’s products. Built upon the framework of the Data Migration Toolkit (see below); this tool serves to decrease the transition period of new customers to SPATIALinfo’s proprietary software suite by flagging specific format incompatibilities and erroneous data. Technologies leveraged include WPF 4, MVVM, Oracle database, WCF.
7. **Data Migration Toolkit** – Implemented automated, extensible, scalable tool to recondition new customer’s existing GIS data into compatible format for consumption by company’s proprietary software suite. Replacing manual, ad hoc processes while decreasing customer transition time. Technologies leveraged include WPF 4, MVVM, Oracle database, WCF.

1. **Web Honesty Website** – Designed, architected and built customer-facing multi-tiered, SQL server-driven website allowing customers to manage account preferences. Emailed, GUIDed URL directs customer to site displaying account preferences. Content includes customized, tabbed, editable AJAX GridView control with minimal client side scripting. This project was originally developed in .NET 3.5 then later deprecated to 2.0 as per business legacy support. Project lifecycle was approximately 3 months. Extensive unit tests and thorough system test cases helped ensure the fast project timeline was met amidst constantly transient requirements. Technologies utilized include C#, .Net framework 3.5, 2.0, AJAX Control Toolkit for ASP.Net 2.0, 3.5, NUnit, SQL Server 2005 stored procedure handling consumption, updating.
2. **Heman Metadata** – Company initiative to replace underlying, legacy infrastructure with “metadata” infrastructure driven by SQL server and automated services.
  - **Active Rolling** – a process maintaining the most active future contract for a financial instrument. This project involved a multi-tiered windowing service (database, application, web service, Unix daemon) that runs multiple times a day in a batch context mirroring market state. The service maintains the current list of active, available and “interesting” (database-defined) future contracts within CQG’s metadata system for broadcast to subscribing client software. Technologies utilized include C#, .Net framework 2.0, windows services, NUnit, web services, SQL Server 2005 database triggers and event handling.
  - **Options Windowing** – Windowing is the process of adding option contracts to CQG’s internal metadata database for consumption by client application software. Options are windowed in via a Windows service that runs in a daily batch context. The rules for which contracts should be windowed are database defined, the currently windowed contracts are highly volatile. This service executes in a multi-tiered context alongside database, a windows service and Unix quote processing daemon all technologies utilized include C#, .Net framework 2.0, windows services, NUnit, SQL Server 2005 database triggers and event handling.
3. **Options Metadata** – Company initiative to drive processing of instruments (financial entities that may be associated with bids) received over data line feeds using pre-existing, large, internal “metadata” system to drive addition, pre-addition of options into the CQG application framework. This initiative spawned many projects.
  - **Auto Add** – This is a parser-based, multi-tiered (parser, web service, database) service that processes exchange metadata to add the financial instruments to CQG’s system in real time. Created web service to interface with UNIX exchange parser for updating internal storage database. Technologies utilized include C#, .Net framework 2.0, NUnit, NCover, web services, XML object serialization and deserialization, SQL Server 2005.
  - **Pre Add** – Pre-Add is a service that creates instruments in CQG's Metadata database before they are needed. Pre-Add only applies to things that expire (e.g. adding the December 2009 Corn Future when the March 2008 Corn Future expires is pre-add). Created four tiered windowing services (database, application, web service, Unix daemon). Extensive automated and manual integration tests were developed for verification. Technologies utilized include C#, .Net framework 2.0, windows services, NUnit, web services, SQL Server 2005 database trigger event handling, and application error reporting.

4. **Stress Testing Framework** – Framework developed to test and determine maximum server load supported by company systems. I developed the message processing system between the C++ client and C# .Net server/data generation service built upon the company's componentized unicast messaging (.Net remoting) implementation. Designed project components, classes in role as project co-architect. Project lifecycle was approximately 2 months. Technologies utilized include Sparx System's Enterprise Architect (UML architecture and documentation application), C#, .Net framework 2.0, windows services, unicast & multicast message processing, XML object serialization/deserialization, NUnit.
5. **Data Factory Replacement** – Initiative to replace the company's data storage framework that retrieves and stores proprietary financial data. Project encapsulated conversion of 20+ year old internal system built in C, C++ to the .Net framework. Initiative canceled 3 months into development as server hardware upgrades running Linux proved further performance gains to be unnecessary (for now!). Technologies utilized include C#, .Net framework 2.0, windows services, SQL database integration via stored procedures, NUnit, NCover.
6. **CQG News** – Developed and deployed CQG News, a componentized application suite handling the storage and transmission of real time news headlines, stories and metadata from financial providers. Joined company & project during its implementation; the subsequent project lifecycle was approximately 7 months.
  - **News Server** –Technologies utilized include C#, .Net framework 2.0, unicast and multicast message processing, windows services, SQL database integration via stored procedures, extensive NUnit unit and integration testing and development.
  - **News Keeper** – Windows service application receiving multicast data from News parsers for database storage. Duties included unit testing, development, integration testing, deployment, support and defect fixing. Technologies utilized include C#, .Net framework 2.0, unicast and multicast message processing, windows services, SQL database integration via stored procedures, extensive NUnit unit and integration testing and development.

Microsoft, Redmond, WA: 2005 –2006

#### **Tool Developer**

1. **WordKey** – A C# desktop tool created for internal use by contextual search advertising analysts within Microsoft AdCenter. WordKey normalizes and deduplicates lists of keywords and ads taken from clients as well as converts from Yahoo! and Google formatting to MSN's own keyword/ad format. Used by teams in France, the UK and USA. Technologies utilized include C#, .Net 1.0, and Microsoft Excel interop.
2. **Batch Invoice Tool** – generates and prints customer invoices to .PDF format. Used by MSN France to generate and print thousands of paper invoices easily. Technologies utilized include C#, .Net 1.0, 3<sup>rd</sup> party proprietary Adobe Acrobat document generation API, XML for document formatting.

3. **AdCenter Desktop** – A C# desktop application built for company executives as revenue auditing tool and key performance alerter. UI-intensive; developed custom Odometer control, map control. Technologies utilized include C#, .Net framework 1.0, 3<sup>rd</sup> party charting API, custom Windows frames, custom controls, Microsoft Excel interop, Microsoft Outlook interop.

America's Growth Capital, Boston, MA: 2004 – 2005

**Tool Developer**

1. **Qwik Note** - accessory tool to simplify company's adoption of online CRM system. Extensive use of third-party web services to manipulate remote database. Technologies utilized include 3<sup>rd</sup> party CRM API (Salesnet), C#, .Net Framework 1.0.
2. **Bloomberg Service** - Windows service to implement real-time Bloomberg market data into newly-adopted Jovus equity research platform via Bloomberg API and third-party TIBCO's proprietary Rendezvous™ messaging system.
3. **DeDuper** - Microsoft Excel plug-in to solve issue of replicated data in company CRM database using .NET and web services. Developed with C# and Excel VB scripting.
4. **Misc Framework** - Proxy class linked library as interface to proprietary 3<sup>rd</sup> party web service API to expedite common application deployment.
5. **Windows Services** - Various Windows services for recurrent retrieval, manipulation and reporting of real-time data via company network and email.