

Ivan Gilchrist
24117 16th Ave SE
Bothell, WA 98021

206 235-9033
ivan@gilly3.com
gilly3.com

EXPERIENCE

Red Violet

Seattle, WA

2/2015 — present

Principal Software Development Engineer

idiCORE.com

idiCORE is a data fusion product which merges datasets from various sources into a massive database, capable of finding precise data very quickly at large scale. I lead the design and implementation of the services and front end layer, focusing on security, performance, and scalability. I headed a team of 4 to 10 developers, interfacing with other technology teams, business owners and stakeholders, executive leadership, and external technology partners. I frequently participated in strategy meetings discussing company goals and product opportunities. I was responsible for interviewing and hiring developers on my own team and participated in interviewing and hiring decisions for positions outside of my team, including technical and executive candidates.

Technologies used: JavaScript, PostgreSQL, C++, AWS, node.js, express.js, vue.js, less

Highlights of my design and implementation:

- Services were designed as a collection of independent worker modules that adhere to a common interface. This enables:
 - Scaling at the worker level. Precise control over scaling out only the workers that require it.
 - Flexible design. Modules could be implemented in different languages or on different platforms if required. This makes A-B testing different implementations very simple.
- One way data flow for security.
- Abstract communication and data storage from worker modules. This makes it easy to swap communication protocols without impacting module code.
- Redundancy for stability.
- I implemented the services in node.js
 - Promises to handle asynchronicity.
 - UDP transport protocol between services.
 - Shared functionality was kept in separate packages, facilitating reuse and cross-team collaboration.
- Node.js add-on, written in C++, to facilitate sharing a UDP transport and queuing load balancer module between systems.
- Data was made available on a customer facing API, utilized by our own web UI, various customers, and our internal batch systems, which accounted for several million searches daily.
- Data was filtered before being delivered to customers, by merging access rules from several sources, including federal and state privacy laws, customer industry rules, and contracts with our data providers. We ensured that no customer received any data it was not allowed to view.
- Authentication system providing authentication tokens from stateless servers, using JWT (JSON Web Tokens). Node.js web server to handle UI login, as well as service authentication, mobile authentication, and internal batch system authentication.
- Authorization system.
 - Authorization rule inheritance, allowing for easy organization, and also fine control of individual exceptions.
 - Effective permissions were included in the authentication token, stored in a proprietary binary format which allowed us to keep the overall token size small, while still including full detail of all permissions.
- Customer database storing relational data for customers, users, passwords, authorization rules, and preferences.
- Administrative tools for customer management, including self-service customer facing tools.
- Web gateway server with outbound, but not inbound, internet access. This was used as a web proxy for our web servers, which were exposed on the internet, but lacked any outbound internet connectivity. This enabled us to isolate systems, narrowing the attack surface and protecting against data breaches.

Lead Software Engineer**Disney Web Services**

Lead the UI team in developing products supporting shared enterprise services.

- Designed and implemented a drop-in JavaScript library (dws.js) abstracting access to authentication, authorization, and other core web services.
- Developed a shared SCSS mixins library, promoting semantic markup and facilitating easy design changes.
- Administrative websites, sample apps, demos, and interactive documentation.
- Provide support integrating services to developers on other teams.

Intellectual Property Registry

A web-based data management system, cataloging Disney's Intellectual Property across the various brands, and managing their relationships. Built on the LAMP stack, with a JavaScript single page architecture using Backbone.js.

- Interfaces with several internal and external REST APIs.
- Elasticsearch DSL queries.
- Client-side file processing using JavaScript File API.
- Responsive design allowing full functionality on mobile devices.
- Mentor junior developers.

Enterprise Readiness Assessment

An operational support tool for collecting data and reporting on the readiness of enterprise systems. It consisted of a MySQL database, a Java Tomcat REST API providing access to the database, a LAMP web app with a heavy JavaScript single page architecture (Backbone, Require, jQuery).

- Principle developer for the full application stack.
- Did a data model re-architecture, which provided a more flexible data model facilitating easier application upgrades in the future. Completed code changes on all application tiers to support the data model changes.
- Developed new functionality, maintaining backward compatibility.

Sr. Software Design Engineer — Office 365 Community Supportability**Agent Dashboard**

A web-based Issue management and Social CRM system hosted on Windows Azure with SQL Azure backend and supporting Windows Azure Worker Role services.

- Primary developer in all application tiers.
- UI and front end:
 - ASP.Net and C# front-end providing rich, dynamic user interaction.
 - JavaScript including extensive AJAX and jQuery, reusable library, jQuery plugins, custom types with inheritance.
 - CSS3 with graceful degradation.
- Middle tier:
 - Robust C# object-oriented design with Object Model and Data Access Layer.
 - Extension methods and library functions
 - Logging infrastructure
- Database:
 - Data modeling
 - Index tuning
 - Stored procedure authoring
- Azure worker role service:
 - Import external data through a REST API.
 - Process business logic through a customizable rules engine.
- Rules engine:
 - Process input against rules to take custom actions based on specified conditions.
 - Translate textual rules into compiled lambda expressions for real-time rule evaluation.
 - Web-based rule builder with free text entry and auto-complete.
 - Object-oriented JavaScript objects to translate UI into parsable rules.

Microsoft (vendor)

Redmond, WA

4/2008 — 10/2010

Software Design Engineer — Health Solutions Group

Microsoft HealthVault Community Connect

An online medical records management system connecting patients to physicians and hospital records.

- Provided extensive design and development in all application tiers: SQL Server, C# data access layer, ASP.Net web front-end, and HTML, CSS, and JavaScript interaction.
- Centralized common functionality into base classes, eliminating redundant code, and making extensibility easier.
- Reduced code complexity and redundancy in data classes with overloads and helper methods using Generics, Linq, and Lambda expressions and delegates.
- Optimized SQL Server queries and views for improved performance and reduced complexity.
- Integrated with Health Vault and Amalga, enabling the cross sharing of patient records and hospital data.

MSN Health & Fitness Route Tracker

Track your workouts as a virtual walk across America or the world. Compete with your friends to finish a walk first.

- Virtual Earth map integration to create custom walking routes and display friends' progress along the route.
- Built a custom JavaScript abstraction layer of Virtual Earth.
- Health Vault integration — Workouts are stored in Health Vault and shared with other Health Vault apps.
- Animated avatar picker.

Provider Directory Search

Interactive, asynchronous JavaScript web control for finding a medical provider.

- Portable — Useable on any website by including a linked script on the page.
- Retrieve search results and result details asynchronously for a quick and seamless user experience.

Expedia.com

Bellevue, WA

3/2004 — 3/2008

Software Design Engineer

Financial Operations — VCard Manager:

- Designed and developed a tool for managing a large, dynamic credit card database.
- This high profile, fast paced project required me to ramp up quickly on several new technologies including: C++, Java, Apache Tomcat, Freemarker, Spring, Java beans, ant, ivy.
- Also used existing expertise in technologies: AJAX, SQL Server, JavaScript, CSS, XML, XSLT.
- UX — Thoughtful and precise use of AJAX technologies to make a fast, intuitive user interface keeping important data available quickly without losing context.
- C++ Legacy tools — Enhanced and integrated existing C++ travel itinerary search and management tools.

Fraud Operations — Aegis Fraud Detection Manager:

- Designed and developed an integrated suite of tools for managing Fraud Detection.
- Technologies — built on ASP.Net, C#, AJAX, JavaScript, XML, XSLT, XPath, SQL Server, Office Web Components; and integrated with a Java web service.
- Rules Engine — Designed and implemented a fast, XML based rules engine, for real-time evaluation of website transactions for detecting fraudulent activity.
- Rules Builder — Designed and built a dynamic UI for building robust, complex fraud detection rules.
- Reporting — Displayed results of recent transactions graphically with dynamic charts and tabular data.

Customer Analytics:

- Designed and developed a reporting site using ASP.NET, XML, XSLT, JavaScript, SQL Server, and OLAP.
- Supported Customer Analytics team statisticians by providing graphical interfaces for manipulating data and viewing custom reports on data including charts, pivot tables, and sortable, searchable grids.
- Developed Windows Services in C# to manage scheduled processes.

Washington Mutual (contracted)

Seattle, WA

1/2004 — 3/2004

Web Developer

Online banking website:

- Accessibility improvements per ADA requirements including making site functional without script or stylesheets.
- Resolved bugs prior to launch, ensuring successful and timely deployment.
- Provided 24x7 production support during product launch by monitoring errors and responding with appropriate corrective action.

- Microsoft (contracted) Redmond, WA 7/2003 — 12/2003
Web Developer
 Developed ASP pages for MSN Subscription signups using ASP, XML, and XSL.
Responsibilities:
- Provide support for international versions of pages for several languages and countries.
 - Responsible for internationalized data validation utilizing xml-based country and language specific data validation rules.
 - Resolve bugs quickly and effectively to hit milestones on time.
- Iron Mountain (contracted) Renton, WA 7/2002 — 6/2003
Web Developer
 Developed front-end for Jaguar, a Billing, Contracts Administration, and Workflow application. Jaguar tracks contracts and handles invoices and billing for thousands of clients. Jaguar is an application used by Sales and Customer Service Reps within DSI, a technology escrow company.
Responsibilities:
- Write ASP pages utilizing VBScript classes.
 - Develop COM Objects in VB for middle tier to interact with front-end asp code and back-end databases.
 - Code Stored Procedures in T-SQL for a SQL Server Database to work with the middle tier and drive the application.
 - Prepare billing invoices in Crystal Reports to support varied invoice formats and contents.
 - Participate in collaborative design sessions for the implementation of complex business rules within the application.
 - Work in a fast-paced, dynamic environment to meet requirements and address new issues as they arise.
- VoiceStream Wireless Bellevue, WA 9/2000 — 6/2002
Web Developer
Streamline:
 A high profile internal application including a knowledge base and transaction processing.
- Enhance functionality and improve site performance.
 - Develop new custom applications.
- Webforia Bellevue, WA 4/2000 — 9/2000
Web Producer
- Created dynamic web pages and web applications using ASP, DHTML, and XML.
 - Deployed and maintained entire web communities.
- Microsoft (contracted) Redmond, WA 11/1999 — 4/2000
Web Developer
- Responsible for sales and marketing team communications on Microsoft's intranet.
 - Developed web pages for members of the sales and marketing team.
 - Used DHTML to build an email based newsletter with international visibility.
- Volt Computer Services Redmond, WA 6/1999 — 11/1999
Administrative Assistant
- Developed JavaScript Rich DHTML games (Jeopardy, Family Feud) for team morale meetings.
 - Managed data for the employee relations team — bringing quality data to the team.
 - Generated reports to be used in meetings with Microsoft.
 - Manipulated data to find trends and analyze performance.

EDUCATION

- Shoreline Community College Shoreline, WA 6/1999
 Studied music, computer science and math.