Samir V Barot (MCSD, Cloud Solution Architect)

41 Kelways Circle, Brampton ON L6T 0G5

C: (416)–277–9950
E: samirbarot@rogers.com

Professional Profile

An experienced and certified professional with extensive expertise in architecting and developing Web, Windows, and native cloud-based applications. Skilled in leveraging Object-Oriented Design (OOD), Service-Oriented Architecture (SOA), RESTful APIs, Event-Driven Microservices Architecture (EDMA). I bring strong architectural and technical proficiency, along with demonstrated leadership and team management capabilities. With excellent verbal and interpersonal communication skills, I excel both as a collaborative team member and as an independent contributor, working effectively with minimal supervision.

Technical Skills Summary

Hardware platforms and Operating Systems

- Dell Latitude, IBM Lenovo, Windows Server 2022/2019/2016, Windows 10/8/7

Languages, Tools and Methodologies

- C#, .NET Aspire, ASP.NET Core 9.x, Microservices, Blazor, Angular, React, VS 2022, TypeScript.NET 8/9, Web API, Microsoft Azure (App Service, Container App, Service Bus, Event Hubs, Cloud Services, Redis Cache, Logic Apps, Key Vault, Configuration, Function App, APIM, Azure Data Factory, AS2 Connector), Dockers, AKS, ASF, ACR, WPF, PRISM 4.0/4.5, SharePoint 2013, Git, GitHub, Office 365 and Microsoft D365
- MS Office 2024 / Office D365, Visio 2019, Project 2019, Rational XDE, UML 2.0, ERwin 3.5, SQL Navigator 4.0
- Waterfall, Iterative-Iteration, Rational Unified Process, Microsoft Solution Framework, Agile, CMMI, Scrum

Databases

- Windows Azure SQL Database, Azure Cosmos DB, SQL Server 2016-22, Oracle 10g, 11i, ADO.NET & EF Core 9.x, LINQ, MS Sync Framework 2.1, SSIS, SSRS, DB2, MongoDB

Work Experience

Accenture/SGI Canada — September 2022 to April 2025

Digital Transformation – DRIVE Auto Fund

Cloud Consultant (SGI Canada)

DRIVE is a modern, cloud-native application built using **Domain-Driven Design** (DDD) principles, developed to support both internal and external issuers at SGI Canada. It empowers users to facilitate a wide range of customer services, including customer management, vehicle registration, auto insurance issuance, driver's licensing services and more. At SGI, I am part of the architectural transformation team for the Auto Fund application. The Mario Kartz team, responsible for the Customer and Vehicle domains, leads this initiative. My role includes leading the adoption of Domain-Driven Design (DDD), implementing an **Event-Driven Microservices Architecture**, facilitating Event Modeling and Storming workshops, delivering technical sessions, and designing and implementing microservices and technical proof of concepts (POCs). This also involves architecting integrations with external systems such as **Insure Cloud, Guidewire, Symcor, SmartComm, OpenText, and CCMTA**. I have been a key contributor across all phases of the software development lifecycle (SDLC), including involvement in UX/UI initiatives, while mentoring the team on various technical aspects such as Azure AI Search, Event Hub, Azure Data Factory, and design patterns throughout the project. To modernize the legacy SAM application, we are leveraging the Strangler Pattern, utilizing both Facelift and Fusion strategies to enable a smooth and incremental migration. Azure Data Factory is also utilized to facilitate data migration from SAM to DRIVE.

Responsibilities:	I was responsible for leading the team in **planning, architecting, designing, developing**, and deploying a **micro-frontend** and **microservices**-based application using React, ASP.NET Core, EF Core, C#, and Azure services such as Azure AI Search, Azure Data Factory (ADF), Event Hub, and API Management (APIM).
Environment:	Windows 10, **VS 2022/ VS Code + TFS**, **Git**, C#, **React, ASP.NET 8/9, EF Core**, LINQ, Web API, Azure SQL Server, Cosmos DB, DevOps, CI/CD Pipeline, Kubernetes
Team Size:	9

NK Techno Labs / Photon Inc — November 2020 to July 2022

API Management Specialist (SME)

Cloud Consultant (AON)

While at AON, I led the architecture and deployment of Azure API Management (APIM) across the organization. This included multi-region deployment, containerized gateways, integration with Okta and ServiceNow, onboarding of backend APIs and Azure Functions via Terraform pipelines, and implementation of various policy levels. I also worked on secure access to Key Vault through policies, along with managing APIs, products, groups, and delegation features.

Responsibilities:	I was **responsible for leading the team, planning, building, and executing architecture, development** and deployment of the **Microservices** using ASP.NET Core/EF Core/C# & and **Azure**
Environment:	Windows 10, **Visual Studio 2022/ VS Code + TFS**, **Git**, C#, EF Core, LINQ, **ASP.NET Core 5/6, Web API**, Azure SQL Server, DevOps, CI/CD Pipeline, API Management, App Services, Azure Functions, Key Vaults, Application Insights, Datadog, TeffaForm and ARM Templates
Team Size:	3

Schedule Optimization

Cloud Consultant (Banfield Petware Hospital)

The Petware application was developed for Banfield clients to support comprehensive pet care—from onboarding and treatment planning to nutrition management and the final exit process. The Schedule Optimization feature enables operators to book client appointments in the most efficient way possible. During the COVID-19 pandemic, Banfield recognized the need to offer Virtual Care and deliver a seamless digital experience that allowed customers to check in their pet at curbside. To support these capabilities, along with real-time scheduling and appointment management, the application was rearchitected using a Microservices Architecture deployed on Azure.

Responsibilities: I am **responsible for architecting and leading the team, planning, building, and executing architecture,** services design, **development** of the **microservices**-based application using ASP.NET Core/EF Core/C# & and **Azure**

Environment: Windows 10, **Visual Studio 2022/ VS Code + TFS**, **Git**, C#, EF Core, LINQ, **ASP.NET Core 5/6, Web API**, **Cosmos DB**, Azure SQL Server, DevOps, CI/CD Pipeline, API Management, App Services, Azure Functions, Service Bus, App Configuration, Key Vaults, Application Insights, TeffaForm and ARM Templates

Team Size: 15

Vantage Integration Layer

Cloud Consultant (UnitedLex)

The Integration Layer was designed to unify various vendor systems—such as **Salesforce, Adobe Sign, Legal Tracker, BriefBox, and Dynamics D365**—through a microservices-based architecture supported by CI/CD pipelines.

Responsibilities: I was **responsible architecting and leading the team, planning, building and executing architecture,** services design, **development** of the **microservices**-based application using ASP.NET Core/EF Core/C# & and **Azure**

Environment: Windows 10, **Visual Studio 2019/ VS Code + TFS**, **Git**, C#, EF Core, LINQ, **ASP.NET Core 3.1/5, Blazor, Web API**, **gRPC**, Azure SQL Server, DevOps, CI/CD Pipeline, API Management, App Services, Azure Functions, Logic Apps, Service Bus, App Configuration, Key Vaults, Moq, SonarCube, Application Insights, TeffaForm and ARM Templates

Team Size: 12

Cineplex **July 2020 to November 2020**

Cineplex Store (Backend)

Consultant

The Cineplex Store backend was developed to support multiple frontend variants—including web, TV, mobile, and Roku—by providing consistent functionality across platforms. It was built using a microservices architecture leveraging Web APIs, Azure Functions, and Azure API Management.

Responsibilities: I was responsible for services and functions design, implementation of the **microservices**-based application using ASP.NET Core/EF Core/C# & and **Azure**

Environment: Windows 10, **Visual Studio 2019 + TFS**, **Git**, C#, EF Core, LINQ, **ASP.NET Core 3.1, Web API**, Azure SQL Server, DevOps, CI/CD Pipeline, API Management, App Services, Azure Functions, Key Vaults, Moq, Application Insights

Team Size: 1

Ceridian / Canada Goose **November 2019 to June 2020**

International Money Movement (IMM)

Consultant (Ceridian)

International Money Movement, a Microsoft Azure Cloud-hosted, microservices-based payroll application, was developed for Ceridian employees to support payroll processing for various clients. The **payroll engine** generated "instructions" (CDD, following the national NACHA standard, or OCC with proprietary formats) for inbound and outbound transactions across **seven banks**. These instructions were uploaded as payroll files to **Azure Blob Storage**, where microservices processed them using business rules, allowing Ceridian employees to take appropriate actions based on client needs. **Fraud and Risk Management** monitored all activities to ensure legitimacy.

The **frontend** was built using **Angular 8** and deployed to **Azure App Service** via **CI/CD pipelines**.
Microservices—covering **Import**, **Transactions and Batch Management** (including Stops, Holds, Reversals, Internal via PSS – Payment Solution Services), **Export** (GL, NACHA, H2H, and Positive Pay), and **Search**—were developed using **Azure Functions** (HTTP and Service Bus Triggers) and **Web APIs**, hosted on **Azure App Service** and exposed through **API Management** (secured with **OpenID Connect** and **JWT authentication tokens**).

For **monitoring, logging, and auditing**, **Application Insights** and **Azure Monitor** were utilized. **Azure Service Bus** (Topics and Subscriptions) enabled communication with external systems such as **CDX** for data exchange.
A robust **CI/CD pipeline** was implemented on Azure to support rapid and reliable deployments, incorporating **unit and integration testing** using the **MOQ framework**. **Microsoft Key Vault** was integrated with both the Function App and Web API for secure **secret management**.
.

Responsibilities: I was **responsible for planning, building and executing architecture,** services design, implementation of the microservices-based application using ASP.NET Core/EF Core/C# & **Angular** and **Azure**

Environment: Windows 10, **Visual Studio 2019/ VS Code + TFS**, **Git**, C#, EF Core, LINQ, **Angular 9**, **HTML**, **Bootstrap, ASP.NET Core 3.1, Web API**, Azure SQL Server, DevOps, CI/CD Pipeline

Team Size: 8

TPL Middleware

Consultant (Canada Goose)

TPL Middleware is a Microsoft Azure Cloud-hosted, **microservices-based application** built for **Canada Goose** partners and third-party logistics providers to process online, employee, and retail orders. It replaces a legacy system and centralizes communication with **security, auditing, logging, state management**, and **message translation** between systems. A **Blazor Server-Side UI** helps the IT Support team trace order issues end-to-end without manually checking **Azure Application Insights** or **SQL logs**. A streamlined **CI/CD pipeline** enables rapid deployments with **unit and integration tests** using **Postman**. The application was built using the following technologies: **Azure Functions** (HTTP & Service Bus Triggers) **Logic Apps** (Routing & Conversion), **ASP.NET Core Web API** with **Swagger, Azure API Management** (with **OAuth**), **Azure Service Bus**, **Microsoft Key Vault**, **Liquid Templates**, and **Integration Account** (D365, Office 365 AS2 for **EDI** with Tigers & True Commerce)

Responsibilities: I was **responsible for planning, building and executing** application architecture, design, implementation of the microservices-based application using ASP.NET Core/EF Core/C# & **Blazor** and **Azure**

Environment: Windows 10, **Visual Studio 2019 + TFS**, **Git**, C#, EF Core, LINQ, **Blazor**, Postman, **HTML**, **ASP.NET Core 3.0, Web API**, Azure SQL Server, DevOps, CI/CD Pipeline

Team Size: 3

PwC — April 2019 to November 2019

Financial Tracker & FxRate — April 2019 to November 2019

Sr. Consultant – PwC

Financial Tracker, **Microsoft Azure Cloud** hosted **Angular SPA** web application was built for PwC to allow it's users to accomplish actual, planned and committed cost and resource planning for all projects PwC. It also generates different reports for the management to help them understand, review, and take critical decisions during or after the planning phase. It covers the full life cycle of the project from initiation to allocation, resource planning and reports. Various imports bring in actual costs from different vendors and backend systems like iPower every month. **FxRate**, an **ASP.NET Core (Blazor Server Side)** based web application was developed for the users of PwC to allow them to calculate conversion rates between different currencies based on historical data (EoD, Monthly and Yearly Average using RESTful API) from Thompson Reuters (Refinitiv). Before starting to work on two applications, I was working on setting up a Microsoft Azure Cloud-based blueprint application which provides the critical guidance on the integration of different technologies like Angular, ASP.NET Core, Setup CI/CD Pipeline (Automation) on Azure, Unit and Integration Tests using Selenium and Sauce Labs, Database Migration, Veracode Static and Dynamic Code Analysis, Integration with PwC Identity, Security, CORS Communication & Key Vault integration. It's finally deployed to the cloud using Web Apps and Docker Containers using ACR. While working at PwC, I received training on multiple Azure technologies like Containers, ASF and AKS from Microsoft, **AWS (.NET Core)**, **UI Path (RPA)**, **Alteryx**, **Tableau** and **Power BI Desktop**.

Responsibilities: I was involved in application architecture, design, implementation of the web-based application using ASP.NET Core/ EF Core/C# & Angular 8/Typescript/HTML/Bootstrap 4

Environment: Windows 10, **Visual Studio 2019 + TFS**, **Git**, C#, EF Core, LINQ, **Angular 8**, **TypeScript**, Postman, **HTML**, **ASP.NET Core 3.0, Web API**, SQL Server 2016, IIS Express 8.0, DevOps, CI/CD Pipeline

Team Size: 7/1

TM7 & RECO — June 2018 to December 2018

C-Quence & Parsley — June 2018 to December 2018

Sr. Consultant – TM7 / RECO (https://www.c-quence.co.uk/)

C-Quence, an Angular based SPA web application, was built for the client of C-Quence to allow them to create commercial insurance (quote/policy) specifically tailored to their needs. It provides search, client eligibility and data validation, auto premium calculations up to 3 options and allows binding preferred options to engage clients directly with a real underwriter. **Parsley**, an ASP.NET Core based web application was developed for the users of RECO to allow them to create different types of applications in **CRM** (based on documents received through background job in **hangfire**) and integrate the generated document with **SharePoint** which was validated for invoice amount using **Spreedly**.

Responsibilities: I was involved in application design, implementation and testing of the web-based application using ASP.NET Core/ EF Core/C# & Angular 6/7/Typescript/HTML/Flexbox/Flex Grid/Hangfire

Environment: Windows 10, **Visual Studio 2017 + TFS**, **Git**, C#, EF Core, LINQ, **Angular 6/7**, **TypeScript**, Postman, **HTML**, **ASP.NET Core 2.2, Web API**, SQL Server 2016, IIS Express 8.0, BitBucket, Jenkins, Source Tree, Hangfire, SharePoint, **Microsoft CRM Dynamics 365**

Team Size: 7/1

Citibank — June 2016 to June 2018

LPM & Amortization — June 2016 to June 2018

Sr. Consultant – Citibank

Loan Portfolio Manager (**LPM**), a web-based SPA application was built for Citibank's internal clients to track the deals from inception through credit approvals including mitigation (CDS, Structures) to ensure the proper booking of corporate Loans, Hedges, CLOs. **The Amortization** application was also built for the clients to get the details on a payment schedule for the corporate loans with RC and RWA calculations. The web-based application was retrieving and persisting the complex data using ASP.NET Core API to the backend database. I have been extensively training with Microsoft Azure cloud for Service Fabric (Microservices and Dockers), App Service (Web Apps), Cloud Services (Web Roles, Worker Roles), ACS, AKS, ACR and Azure Storage (Blob Storage)

Responsibilities: I was involved in application design, implementation and testing of the web-based application using ASP.NET MVC, ASP.NET Core /C#/ Angular 2/5/Typescript/HTML/Entity Framework

Environment: Windows 7, **Visual Studio 2017 + TFS**, **Git**, .NET Core, C#, EF 7.0, LINQ, **Angular 2/5**, **TypeScript**, Postman, **HTML**, **ASP.NET Core,** SQL Server 2016, IIS 8.0

Team Size: 6

Allianz Global Assistance — May 2015 to April 2016

Claims Portal — December 2015 to April 2016

Sr. Consultant – Allianz

Claims Portal, a web-based SPA application was built for Allianz to create any kind of quote (Technical – TC, TI, FLD or Medical – HCFA, UB) for its customers. The web-based application was persisting the complex data using ASP.NET Core API to Claims Database. The task scheduler and iSynergy ASP.NET Core API were used to schedule to submit data to the backed DB2 system and update the case number and other statuses back to Claims Database. This web application was built using ASP.NET Core 1.0 (RC1), C#, Angular 2 (Beta) and TypeScript, HTML, CSS3, Bootstrap, Entity Framework. A prototype was also built to allow claims data to be persisted in the MongoDB database.

Responsibilities: I was involved in application architecture, design, implementation and testing of the web-based application using ASP.NET Core 1.0/C#/ Angular 2 (Beta)/Typescript/HTML/Entity Framework

Environment: Windows 7, **Visual Studio 2015 + TFS**, **Git**, .NET 4.6/Core, C#, EF 7.0, LINQ, **Angular 2 (Beta)**, **TypeScript**, Postman, AutoFac, Mock, **HTML**, **ASP.NET Core** 1.0, SQL Server 2014, IIS 8.0, DB2, **MongoDB**

Team Size: 3

Travel Medical Insurance (TMI) Application (TD Bank) — May 2015 to November 2015

Sr. Consultant – Allianz

Travel Medical Insurance, a web-based application was built for TD Bank to get the quote for travel insurance for its client. It was used by internal staff members as well as customers of TD Bank. The web-based application was collaborating with the complex and comprehensive back-end processing logic using WEB API to allow the user to select different plans while filling the entire application. This website was built using ASP.NET MVC, C#, HTML, CSS3, Bootstrap, jQuery, WEB API, Entity Framework. I also spent extra time developing a prototype for an intranet web application using Angular and TypeScript and ASP.NET Core 1.0 to show the usage of different technologies for future in-house and client projects. (https://www.tdcanadatrust.com/products-services/banking/travel-medical-insurance/#)

Responsibilities: I was **leading the team** and also involved in application design, implementation and testing of the web-based application using ASP.NET MVC/C#/HTML/jQuery/WEB API/Entity Framework

Environment: Windows 7, **Visual Studio 2015 + TFS**, **Git**, .NET 4.5, C#, EF 6.0, LINQ, **Angular**, TypeScript, Postman, AutoFac, Mock, **HTML**, **ASP.NET MVC** 5.0, Web API, SQL Server 2014, IIS 8.0 and SharePoint 2013

Team Size: 3

Systemgroup Consulting — September 2014 to January 2015

PlanLink Quote Application (Manulife Financial) — September 2014 to January 2015

Sr. Consultant – Manulife

PlanLink, a web-based application, was built to get the quote for insurance for its client. It was used by internal staff members of the Manulife Financial Group. The web-based application collaborated with the complex and comprehensive rule engine using a WCF service to allow the user to have enough information or guidelines while filling in the entire application. This website was built using ASP.NET MVC, C#, HTML, Bootstrap, WCF, WEB API, EF and Angular JS.

Responsibilities: I was involved in application design, implementation and testing of the web-based application and rule engine using ASP.NET MVC/C#/HTML/jQuery/WEB API/WCF/Entity Framework/Angular

Environment: Windows 8.1, **Visual Studio 2013 + TFS**, .NET 4.5, C#, Entity Framework 6.0, LINQ, **Angular 1.2**, **HTML**, **ASP.NET MVC** 5.0, Web API, WCF 4.5, SQL Server 2012, IIS 8.0

Team Size: 7

CIBC Wholesale Banking Technology (Capital Markets) — July 2014 to September 2014

London Data Warehouse (LDW) — July 2014 to September 2014

Sr. Consultant – CIBC

LDW, a flexible tool, was built to store customer, trade and position data and general ledger balances for regulatory reporting and compliance purposes. It was used by internal staff members of the CIBC Wholesale Banking Technology or Capital Markets group (WBT). The website was communicating with the complex backend system using WEB API to get the display and update trade and customer-related data. This website was built using ASP.NET MVC, C#, HTML, CSS3, Bootstrap, jQuery, Web API, Entity Framework, jqGrid, KnockoutJS, Moment JS, and SharePoint.

Responsibilities: I was involved in application design, implementation and testing of the website and REST services using ASP.NET MVC/C#/HTML/jQuery/Web API/Entity Framework/Knockout JS

Environment: Windows 7, **Visual Studio 2013 + TFS**, .NET 4.5, C#, Entity Framework 6.0, LINQ, jqGrid, **Knockout JS**, Moment JS, **HTML**, **ASP.NET MVC** 5.0, Web API 2.0, SQL Server 2008, IIS 7.0 and SharePoint 2013

Team Size: 4

Infrastructure Ontario — March 2014 to June 2014

Webloans — March 2014 to June 2014

Sr. Consultant – Infrastructure Ontario

Webloans, a service-oriented client-facing website, was built for the different organizational clients to apply for the loan for the various infrastructural projects they initiated during the year. It was also used by internal staff members to apply for a loan on behalf of the organizational client. The internal application also supports the maintenance of database tables. The website was communicating with the backend system named Portfolio Plus using WCF service to get organizational information and to submit the loan data so that the backend system can be processed further. This website was built using ASP.NET MVC, C#, HTML, CSS3, Bootstrap, jQuery, Web API, WCF Service, Entity Framework, Http Client, and Unity.

Responsibilities: I was involved in application architecture, database design, implementation and testing of the web site, REST services using ASP.NET MVC/C#/HTML/jQuery/Web API/WCF/Entity Framework

Environment: Windows Server 2012, Windows 7, **Visual Studio 2013 + TFS**, .NET 4.5, C#, Entity Framework 6.0, LINQ, POCO, SQL Server 2012, IIS 7.0, **HTML**, **ASP.NET MVC** 5.0, Web API 2.0, CSS 3

Team Size: 1

Morneau Shepell — May 2013 to December 2013

Web Portal — May 2013 to December 2013

Sr. Consultant – Morneau Shepell

The Web Portal was multitenant (where single IIS instance was responsible for rendering every client), a service-oriented client-facing website where the user can access information from multiple lines of business. It was used by the clients of Morneau Shepell for different services like Human Resources Consulting, Employee, and Family Assistance Program, Pension and Benefits Plan. This website was built using ASP.NET MVC, C#, HTML, CSS3, jQuery, Web API, Entity Framework and Spring.NET. One tool of the web portal was developed using AngularJS.

Responsibilities: I was involved in application and database design, implementation and testing of the website, REST services using ASP.NET MVC/C#/HTML/JQuery/Web API/Entity Framework

Environment: Windows Server 2012, Windows 7, **Visual Studio 2012/2013 + TFS**, .NET 4.0/4.5, C#, Entity Framework 5.0, LINQ, POCO, SQL Server 2012, IIS 7.0, **HTML**, **ASP.NET MVC** 4.0, Web API, CSS 3, AngularJS

Team Size: 4

Agricorp — July 2011 to March 2013

RMP Livestock (Risk Management Program – Livestock) — July 2011 to March 2013

Sr. Consultant – Agricorp

RMP Livestock application was a Risk Management Program designed and used by the Ontario Ministry of Agriculture, Food and Rural Affairs (OMAFRA) in collaboration with Agricorp. This application helps producers (of cattle, hogs, sheep, and veal) manage risks beyond their control, like fluctuating costs and market prices. The RMP Livestock program worked like insurance where producers select their coverage level, pay premiums and receive payment if the average market prices for enrolled livestock fall below the support level. The application was built using WPF, PRISM, WCF and Entity Framework.

Responsibilities: I was involved in application and database design, implementation and testing of the smart client application, web services using C#/WPF/PRISM/Entity Framework

Environment: Windows Server 2008 R2, Windows 7, **Visual Studio 2010/2012 + TFS**, .NET 4.0/4.5, C#, Entity Framework 4.0, LINQ, POCO, SQL Server 2008 R2, IIS 7.0, **WPF**, **PRISM** 4.0, WCF 4.0, Telerik Controls, Oracle 10g, Reactive Extensions 2.0, Pex and Moles, Ghost Doc, Design Patterns

Team Size: 7

MCSCS (Ministry of Community Safety and Correctional Services) — December 2009 to July 2011

LSI-OR (Level of Service Inventory – Ontario Revision) — December 2009 to July 2011

Sr. Consultant - Ministry of Community Safety and Correctional Services

LSI-OR, a Smart Client Application was a risk/need assessment tool used by the MCSCS in Ontario. This tool enabled a classifier to determine an offender's likelihood of re-offending while balancing the offender's programming needs. The LSI-OR contains dynamic indicators that reflect changes in the offender's score as the offender's circumstances change. The LSI-OR determines this probability based upon an examination of the criminogenic factors present in an offender's life. LSI-OR was linked to the ministry's Offender Tracking Information System (OTIS). The application was built using the Occasionally Connected System Architecture using WPF, PRISM, Web Services, MS Sync Framework and Entity Framework. A Management Website was also built using ASP.NET MVC Dynamic Data Website and Entity Framework and LINQ.

Responsibilities: I was involved in application and database design, implementation and testing of the smart client and web application, web services using C#/WPF/PRISM/ASP.NET MVC and Dynamic Data

Environment: Windows Server 2008 R2, Windows 7, **Visual Studio 2010 + TFS**, .NET Framework 4.0, C#, **ASP.NET MVC** 2.0/3.0, Dynamic Data 4.0, Entity Framework 4.0, LINQ, POCO, SQL Server 2008 R2, SQL CE 3.5 SP2, IIS 7.0, MS Sync Framework 2.1, **WPF**, **PRISM** 4.0, WCF 4.0, Telerik Controls, Oracle 10g, Enterprise Library 5.0, Click Once, Unity 2.0, CAS, WAS, Pex and Moles, Ghost Doc, Software Factories (WCFS, WSSF), Design Patterns

Team Size: 6

ObjectSharp Consulting & AllStream & Gore Mutual — July 2008 to December 2009

Sr. Consultant

While staying at ObjectSharp & Allstream, I was working for their clients using .NET technologies. While working at Gore Mutual, I was working on designing and implementing a centralized CIF Database, Client Broker Services (WCF Service and ADO.NET Data Services) to allow Duck Creek application to access the database through Entity Framework. While working at TD Bank, I was responsible for application and database design and development. I have created an application to automate the generation and publication of the FX Wash Report from backend data to the SharePoint site in the form of PDF.

Responsibilities: I was responsible for application and database design, implementation and testing of the front end, web services using C#/ASP.NET.

Environment: Windows Server 2008, Windows 7, **VSTS 2008/2010 Beta 1**, .NET Framework 3.5/4.0, C#, ASP.NET & MVC, WCF, WF, SharePoint, LINQ, Silverlight, SQL Server 2008, IIS 7.0

Team Size: 3

MCI (Ministry of Citizenship & Immigration) — February 2008 to June 2008

eNominations — February 2008 to June 2008

Sr. Consultant - Ministry of Citizenship and Immigration - https://www.appenom.citizenship.gov.on.ca/enom/

eNominations, a web-based application was built for the people of Ontario to electronically file nominations all year round from any location. The main purpose of this application was to provide end to end web-based solutions to automate and improve customer service and to promote volunteerism and provide greater awareness of recognition programs in Ontario. A Smart Client application was also built to streamline and improve processing time for OHAS (Ontario Honours and Awards Secretariat) staff by enabling automatic data transfer using web services.

Responsibilities: I was responsible for leading the team, architecture, application and database design, implementation and testing of the front end, web services and smart client using C#/ASP.NET.

Environment: Windows 7, **VS 2008**, .NET Framework 2.0/3.5, C#, ASP.NET, SQL Server 2005, IIS 6.0

Team Size: 2

OACCAC (Community Care Access Centres) — **March 2007 to November 2007**

Common Case Management (CCM) — **March 2007 to November 2007**

Sr. Consultant - OACCAC

Common Case Management, an enterprise-class web application was built for the case managers of CCAC. The main purpose of this application was to provide end-to-end web-based solutions to automate and improve the community care access services to the clients. This application was built using an n-tier architecture with implementations of custom (interactive and container) controls, entities, business objects, mappers and factories and services (kind of **CSLA.NET**) which in turn integrate with HPG (Health Partner Gateway). While working on this project, I was also learning/practicing **MOSS 2007/WSS 3.0** in the form of a webcast, white papers and attending user group meetings.

Responsibilities: I was responsible for design/implementing/testing the client-side and administration of the CCM system using C#/ASP.NET.

Environment: Windows Server 2003, **Visual Studio 2005**, .NET Framework 2.0/3.0, C#, ASP.NET, SQL Server 2005, IIS 6.0, BizTalk Server 2006, Telerik AJAX Controls

Team Size: 5

Elections Ontario — **September 2006 to March 2007**

Election Management System (EMS) — **September 2006 to March 2007**

Sr. Consultant - Elections Ontario

Election Management System, an enterprise-class web application was built for the staff of Elections Ontario. The main purpose of this application was to provide end to end web-based solutions to automate, improve and accelerate and error-prone the **Election Management Process**. There were many modules like Event, Location, Ballot, Elector, Field, Material & Budget and HRIS Management. The purpose of the M & B module was to centralize a paper-based process and to stay in sync with the central budget. The purpose of the HRIS module was to centralize the paper-based employee management (lots of paperwork because of short-term employees & also GP doesn't provide anything for managing positions and applications) and providing access to field users to fill out the information about applicant & position instead of finance staff. This application was built using an n-tier architecture with implementations of custom (render, interactive and container) controls, entities and services which in turn accessing the MS Dynamics web services for Finance and HRIS module.

Responsibilities: I was responsible for design/implementing/testing the Material/Budget and HRIS module of the EMS system using C#/ASP.NET.

Environment: Windows Server 2003, **Visual Studio Team System (Developer Edition)**, .NET Framework 2.0, C#, ASP.NET, SQL Server 2005, IIS 6.0, Dynamics GP (Financial, HR)

Team Size: 5

BMO — **February 2006 to August 2006**

Customer Connect System — **February 2006 to August 2006**

Sr. Consultant - BMO & Microsoft

Customer Connect System, an enterprise-class Composite UI Application was built for the tellers of Bank of Montreal. The main reason for building this teller application was to have a single shell application to load different modules as required so that the teller can do different activities like Account, Mortgage, Insurance and other bank-related activities through a single shell without switching between different application. This application was built using n-tier architecture varying implementations of BMO Framework, BMO Controls, Service Abstraction Layer, Application Framework, Presentation Business and Data Layers and their serialization across communication tiers. BMO & Application Framework was implemented using Reflection and different design patterns like Composite, Dependency Injection, MVC, MVP and Domain Model. Attribute and Interface-based programming were used to build the different services provided by BMO Framework.

Responsibilities: I was responsible for implementing/testing the BMO Framework using C#.

Environment: Windows Server 2003, **VSTS (Developer Edition)**, .NET Framework 2.0, C#, Windows Forms

Team Size: 12

Procom (NIAD) — **January 2006 to February 2006**

Virtual Syndication Management — **January 2006 to February 2006**

Sr. Consultant - CIBC World Markets

Virtual Syndication Management was implemented to streamline the secure new issue sales process for brokerage firms like CIBC WM, for which I developed the client POC.

Responsibilities: I was responsible for designing /implementing/testing the front, middle and data layer using C#/ASP.NET, MS Reporting Service, and SQL Server Analysis Services.

Environment: Windows 7, VS.NET 2005, .NET 2.0, C#, ASP.NET, SQL Server 2005, SSRS, SSAS, SSIS

Team Size: 1

Ministry Of Transportation (MTO) — **April 2005 to September 2005**

Claims Information Management Systems (CIMS) — **April 2005 to September 2005**

Sr. Consultant (Technical Architect – Level 2) - Ministry of Transportation (MTO)

CIMS, **Claims Information Management Systems** was considered and developed for the people of collection department to provide the ability to manage any damage claim occurred on government property as a part of an accident/any event which includes generating a claim, repair report, fire department report invoice, billing request, deposit receipts and administration by different users. This application was built using n-tier architecture varying implementations of Presentation, Facades, and States, Business, and Data Layers and their serialization across communication tiers**.** The front-end was implemented using **MVC** including page controller implementations. The middle-tier was implemented using Web Services**,** Facades, Business Services and Entities **were implemented with an industry-standard pattern** such as singleton, adaptor, facades, proxy and more. Database-tier was also implemented using helper and library classes.

Responsibilities: I was responsible for designing/modelling/implementing/testing the front (presentation facades, state), middle (business services and entities), web services and data tier using C#/ASP.NET.

Environment: Windows XP Professional, Windows Server 2003, VS.NET 2003, .NET Framework 1.1, C#, ASP.NET, Oracle 9i, ADO.NET, XML, Rational XDE, UML, ClearQuest, **RUP, CASE Tools**

Team Size: **2**

Susquehanna Technologies — January 2005 to February 2005

Air Force Non-appropriated Fund Purchasing Office. — January 2005 to February 2005

Sr. Consultant - Susquehanna Technologies/Brainhunter.COM

IBPS, an **Internet-Based Purchasing System** (**eProcurement**) was designed and developed to provide purchasing and contract management support for Air Force activities that operate using Non-appropriated Funds (NAF).

Responsibilities: I was responsible for implementing/testing the front, middle and data-tier using VB/ASP.NET and Business Objects using **CSLA.NET** architecture.

Environment: Windows XP Professional, VS.NET 2003, .NET Framework 1.1, VB.NET, SQL Server 2000, ADO.NET, CSLA.NET

Team Size: **15**

Sunlife Financial — November 2004 to January 2005

Rates Automation — November 2004 to January 2005

Sr. Consultant - Sunlife Financial

Rates Automation, Web Application, was built for admin users of ABC/SLF websites of Sunlife Financial. The validation of this application was to adjust the rates for different products offered by Sunlife.

Responsibilities: I was responsible for implementing/testing the front, middle and data-tier using VB. NET/ASP.NET

Environment: Windows 2000, Visual Studio 2003, .NET Framework 1.1, VB.NET, SQL Server 2000, ADO.NET

Team Size: **4**

Royal Bank of Canada — February 2004 to October 2004

PIP (Personal Investment Profiler) — February 2004 to October 2004

Sr. Consultant - Royal Bank of Canada (RBC Investments)

Personal Investment Profiler (PIP), Smart Client Application, was built for Investment Counselors (IC) and Investment Advisor (IA) of high net worth department of RBC Investments. The rationalization of this application was to introduce a client discovery solution which uses interactively with a client to collect personal information (KYC), analyze a client's risk tolerance, determine the investment model using logic-based scoring the client most closely identifies with and recommends a set of products based on the investment model. It had capabilities of working online/offline mode and also had a release management strategy built-in for any future releases of it.

Responsibilities: I was responsible for designing/implementing/testing the front, middle and data-tier using C#. NET

Environment: Windows 2000 Professional, Visual Studio.NET 2003, .NET Framework 1.1, C#.NET, SQL Server 2000, ADO.NET, **Infragistics** Third Party Controls, Windows Forms

Team Size: **5**

OTPP (Ontario Teachers' Pension Plan) — December 2003 to February 2004

Application Framework Management Tool — December 2003 to February 2004

Sr. Consultant - OTPP (Ontario Teachers' Pension Plan)

Application Framework Management Tool, a thick client app, was built for OTPP developers to manage application definitions without direct database access**.**

Responsibilities: I was responsible for designing/implementing/testing the front, middle using C#. NET

Environment: Windows XP Professional, Visual Studio.NET 2003, .NET Framework 1.1, ASP.NET, C#. NET, SQL Server 2000, ADO.NET

Team Size: **1**

York University — April 2003 to August 2003

Violation Payments and Permit Renewals Application — April 2003 to August 2003

Sr. Consultant - York University

Violation Payments and Permit Renewals web-based Application was built for all the students of York University. The rationale of this application was to allow the students to go online and check/pay the violations on their car(s) as well as allow them to renew the parking permit for the next term/semester/year. The front-end was implemented using **ASP.NET**/**VB.NET** along with the use of **XML/XSL** for rendering some of the contents. The middle-tier was implemented using **Web Services (VB.NET)** and XML content was passed between three tiers. Database-tier was also implemented using helper and library classes (ADO.NET). The deployment was done on **three physical servers** where middle-tier and database-tier were deployed using **clustering** and frond-end was deployed using **load balancing** techniques.

Responsibilities: I was responsible for implementing/testing/deploying the front-end (ASP.NET/VB.NET), middle-tier (**Web Services**) and back-end (Helper, Library Class/ADO.NET) systems.

Environment: Pentium IV 1.0 GHz, Windows XP Professional, **Windows.NET Server 2003 (Beta)** and Visual Studio. NET, .NET Framework 1.0, ASP.NET, VB. NET, Web Services, Microsoft XML DOM, XSL Style sheet, SQL Server 2000, ADO.NET, IIS 5.0 & 6.0

Team Size: **2**

Ernst & Young — February 2003 to April 2003

GFIS – Global Financial Information Systems — February 2003 to April 2003

Sr. Consultant - Ernst & Young

GFIS, Global Financial Information Systems, the web-based application was built for all the users of Ernst & Young Global (EY Global). The purpose of this **GFIS** application was to move all the EY Suite Applications (Mainframe and Lotus Notes based) to the global level using new design and implementation.

Responsibilities: I was responsible for developing the front-end, middle-tier (VB.NET/C#.NET) and data-tier (ADO.NET). I was also responsible for Meeting Room Booking and EY Phone Book systems.

Environment: Windows 2000 Server, Visual studio.NET, .NET Framework 1.0, ASP.NET, VB.NET, C#. NET, Microsoft XML DOM, SQL Server 2000, ADO.NET, IIS 5.0

Team Size: **8**

Manulife Financial **July 2002 to Nov 2002**

eTreasury Automation **July 2002 to Nov 2002**

Sr. Consultant - **Manulife Financial**

eTreasury Automation web-based application was built for the users from the eTreasury department. The purpose of this eTreasury Automation application was to design and implement the manual EFT/CHEQUE requisition functions of the eTreasury department. Different functionality like, creating/editing templates and requisition, security user, approver user and exports with backend system and releasing of requisition were designed with **ASP.NET**, which was interacting with **VB.NET** components and **ADO.NET** components. The use of XSL to render the XML content to HTML separates the logic from the data.

Responsibilities: I was responsible for analysis/design/implementation/testing for all tiers of an application. I was also **leading the team** of two people from a technical and management point of view.

Environment: Windows XP, .NET Framework 1.0, ASP.NET and Visual Basic. NET, Microsoft XML DOM, XSL Style sheet, SQL Server 2000, ADO.NET, IIS 5.0, Visual Studio.NET

Team Size: 3

Xwave **March 2001 to March 2002**

Sr. Software Consultant

While working at Xwave, I was responsible for the analysis, design, implementation, testing, and documentation of applications. During my stay at Xwave, I had developed the Xwave Intranet, Kingsway Point of Sale (POS) applications.

Responsibilities: I was responsible for analysis, design, implementation, testing of applications and documentation using Microsoft Technologies.

Environment: Windows 2000 Server, Visual Studio. NET Beta 2, .NET Framework 1.0, ASP.NET, VB.NET, C#, Microsoft XML DOM, XSL, SQL Server 2000, ADO.NET, Microsoft SharePoint Portal Server 2001, IIS 5.0, Microsoft XML DOM, XSL Style sheet

Team Size: 2

North American Media Engines (NAME) **July 2000 to February 2001**

Consultant – Team Leader

While working at North American Media Engines (NAME), I was responsible for analysis, design, implementation, testing, and documentation of applications.

Responsibilities: Involved in application/database design/analysis/implementation/testing and documentation of the software. I was also **leading the team** from the Technical Management point of view

Environment: Windows NT 4.0, Story Server 5.0, Tcl 3.0, 5.0, SQL Server 7.0, NEO Object Library

Team Size: 4-13

Cygnus Computer Associates Ltd. (Teletech Inc.) **April 1998 to July 2000**

Sr. Technical Systems Analyst

While working at Cygnus Computer Associates Ltd (Teletech Inc.), I was responsible for the analysis, design, implementation, testing, and documentation of applications. During my stay at Cygnus, I had developed Bell Direct Services - Order & Status (BDSO&S), Personal Address System (PA system), Bell Atlantic North Reporting System (Team Leader), Cybercare (Team Leader) and Pike – Order Tracking System (Team Leader). I was mostly using MS technology to build these applications.

Responsibilities: Involved me in application/database design/analysis/implementation/testing of the software. I was also **leading the team** from the Technical Management point of view.

Environment: Windows NT 4.0, VB 6.0, ASP, MTS, XML and Oracle 8.0, IIS 4.0, SQL Server 7.0, MSMQ, COM, COM+

Team Size: 4-5

Royal Bank of Canada **October 1997 to April 1998**

Technical Systems Analyst

Electronic Business Banking (EBB)

Responsibilities: Took part in redesign, implementation and testing the software from a 16-bit platform to a 32-bit platform. I was working on the Platform part of the application.

Environment: IBM PC, Windows' 95, Visual Basic 5.0 and Microsoft Access 97

Team Size: 9

Education

Gujarat University, **1990** - **1994**, India
Bachelor of Engineering (B.E.) in Computer Engineering, DDIT, INDIA
Honour: First Class with Distinction