Curriculum Vitæ

Stewart Gebbie

January 4, 2016

Personal Details

Full Name: Stewart Gebbie

Date of Birth: 23rd May 1975

Marital Status: Married

Nationality: South African

South African Identity Number: 750523 5096 08 9

Contact Details

Telephone Numbers:
(Home Phone) +27 11 678 8470,
(Cell Phone) +27 84 738 2899

E-Mail Address: sgebbie@gethos.net

Postal Address:
P.O.Box 215
Glenvista
2058
Johannesburg
South Africa

Contents

1 Technical Summary

I have been involved in a variety of projects ranging across small utilities, systems integration, distributed systems development and software engineering related consulting. These have targeted various programming languages, operating systems and environments.

Being involved in the full project life-cycle it is important to ensure that the problem domain is well defined and appropriately scoped. This is supported by the necessary requirements gathering and subsequent investigations and analysis so as to arrive at a clear understanding of the problem. Based on this, an architecture and design can be formulated that incorporates the necessary modularity, appropriate design patterns, and support for data flows as needed in order to solve the problem and acknowledge the inherent commonalities and variabilities.

Many of the systems I have built are highly concurrent in nature and may additionally depend on distributed interactions. In order to ensure suitable quality and maintainability, automated unit testing and load testing is required, generally developed following a test first methodology.

The final product is the result of iterative refinement and refactoring. This requires ongoing review of all levels of design, from implementation coding, to architecture, and back to the ratification of the original scope and assumptions. Thus ensuring a deliverable of high quality that addresses the requirements of the problem being solved.

Principle Skills

systems architecture, systems design, systems integration, non-abstract large scale systems design, concurrency, event-driven systems, actor model, distributed systems, asynchronous messaging, network programming, unit testing, Java, C, C++, UNIX/Linux.

2 Objectives

I would like to work on projects that involve large scale distributed systems while providing interesting and challenging problems that will draw on my skills as a software engineer and the formal thinking gained via my background in pure mathematics.

3 Education

Tertiary Postgraduate (1999-2001) MSc Mathematics (1999-2001, Degree awarded in May 2002)

Thesis: A Survey of the Mathematics of Cryptography (78% / A) (Accepted without changes following favourable reports from internal and external examiners) The written component covers the history of **cryptography** up to modern ciphers and also includes modern **encryption** techniques (e.g. block ciphers, public key encryption, digital signatures) and **cryptanalytic** techniques (e.g. differential cryptanalysis, linear cryptanalysis). In addition to the written component I also developed **C++** implementations of DES and programs necessary for performing differential cryptanalysis of DES.

Course work: (66% / C) Analysis of Algorithms, Combinatorics, Number Theory, Stochastic Calculus.

Tertiary Undergraduate (1993-1996) *Studies at The University of the Witwatersrand (Johannesburg, South Africa).*

BSc Honours Mathematics (65% / C) (1996)

BSc with Mathematics (74% / B) and
Computational Applied Mathematics (76% / A) Majors (1995)

Additional Courses:

Pattern Oriented Software Architectures for Concurrent and Networked Software – Coursera – February 2013
Intel Multi-core Programming Workshop – Intel – August 2009
The South Africa Grid School – July 2008

Scholarship/Bursary

1993-1995 Bursary for BSc Undergraduate studies from Liberty Life.
1993 University Council Entrance Scholarship
1994 University Council Merit Scholarship

Prizes/Awards

1993 Certificate of First Class - Computational & Applied Mathematics I
1993 Certificate of First Class - Computer Science I
1993 Certificate of First Class - Mathematics I (Major)
1993 Certificate of Merit - Physics I (Major)
1994 Certificate of First Class - Computational & Applied Mathematics II
1994 Certificate of First Class - Mathematics II
1995 Certificate of First Class - Computational & Applied Mathematics III

Secondary:

Mondeor High School. (Johannesburg, South Africa)
Matriculated (1992) with 5 distinctions.
Computer Science - A, Mathematics - A, Science - A, Biology - A, Geography - A, Additional Mathematics - C, English - C, Afrikaans - C.

4 Jobs

4.1 Google

April 2014 to Present

Working in Zürich as a Site Reliability Engineer for Google Switzerland GmbH.

I am part of a small team that supports and maintains production aspects Google Docs and Google Drive. This requires understanding a large part of the internal stack at Google that enables systems to be developed and deployed at scale. The direct systems span multiple data centres and rely on global infrastructure and routing mechanisms.

This has added to my skills and given my key insights into the tasks required to run distributed systems at scale: design patterns for global systems, release management, telemetry, monitoring and alerting, capacity planning.

Additionally, I have been integral to an internal project, Compass, which is aimed at introducing a global new request routing layer for Google Docs to improve the handling of requests that require task affinity during processing.

4.2 Gethos

March 2005 to March 2014

I am the founder of Gethos which I have run independently providing systems development skills and consulting to clients. The projects have been varied in nature: cross-platform development, performance optimisation, systems integration, development life-cycle, systems architecture.

In addition to client projects, there I have run internal projects to build tools needed for the work I have provided to my clients.

Internal Projects:

1. Cohesion *October 2009 to Present*

   I have developed Cohesion as a lightweight, high performance and resilient **actor** based **framework** for **Java** that enables building of **high concurrency** distributed systems while simultaneously facilitating system wide testing. This enables high concurrency even with heterogeneous work loads with actors interacting via **asynchronous** messaging for building **event driven** systems.

   The key areas in this include: **lock-free** workers, **loose coupling** through structural typing, support for different message representations, btree persistence, **non-blocking** TCP and UDP I/O, **multiplexed streaming** over TCP, **dependency injection**. All of these have been developed exclusively by myself.

   The framework encourages a "simulation first" programming style, whereby simulations of the complete system backend can be instantiated without concern for **remote** nodes. Once behavioural correctness is established then the system can be targeted at multiple remote compute nodes.

Frontend integration is supported via seamless mapping of **asynchronously** communicated **JSON** to internal binary representations of POJOs, together with an embedded web server.

2. SEDA Util *circa 2008*

   I developed this Java **framework** to provide facilities for building **Staged Event-Driven Architecture** based systems. Stages are connected via messages queues, and each **stage** maintains its own thread pool and performs a specific task. This enables high **concurrency** when work associated with a given message needs to move through a pipeline. Stages can also implement scatter/gather semantics with the help of **barriers**.

   This framework has enabled the development of high throughput processing while providing integration between disparate systems.

   Provided to clients under GPL.

3. CUnit *circa 2004-2007*

   I developed CUnit as 'C' XUnit testing **framework**. This was developed to simplify **unit test** development for cross-platform C projects. The key feature was that each test suite was complied into a separate shared library which the test runner would then load dynamically.

   Released as open source under GPL.

Client Projects:

1. ECN/Nashua Communications: *various projects from 2012-2013*

   Nashua Communications provides **VoIP** services. More specifically it started providing least-cost-routing by moving voice calls from the POTS network onto a **SIP** based VoIP network to reduce the cost of transmission between parties nationally (in South Africa) and then break out back into the POTS network.

   I was first involved in a consultative roll to help introduce software development **best practices**: continuous integration, test first programming, improved use of version control etc.

   Following this, I provided architectural input and developed various subsystems.

   Some utilities included: AMI event streaming, HTTP forwarding, **StatsD** integration etc).

   Distributed systems that I architected and implemented included a **realtime billing** system and a query network. The realtime billing system leveraged the actor model to build a **sensor network** with error budgets to manage realtime billing of calls across the distributed system. Complexities in the error budget algorithm saw this being successfully replaced by a simpler realtime billing aggregation mechanism.

   Finally, a new "services" network has been architected in order to allow for more variability in the types of voice services that Nashua Communications wishes to offer in the future. This, again, relies on the **actor model** as the underlying execution pattern. In order to support distributed configuration of the system a content addressing approach has been taken. The dissemination of service configuration data is performed using an event driven model (which again can be cleanly built on top of the actor model). This design should allow new configurations to be rolled out to the network while older configuration are still active and not yet retired.

2. Cibecs: *various projects from 2007-2013*

   Cibecs provides enterprise desktop backup via systems primarily developed in Java. The underlying mechanism uses the **Rsync** algorithm. This produces **deltas** that are recorded on the backup

server, thus providing incremental backup. In order to restore from backup, however, a large number of deltas need to be merged in order to recreate a single effective delta. This process was producing very large merged deltas and was slow. I **optimised** the delta merge process by creating internal data structures that matched the delta literal and copy **command patterns**. I further optimised the process by using the **smallest primitive types** needed to represent a given command (e.g. int vs long). The combination of these massively **reduced** the **memory** requirements for the process and in turn significantly **improved** the **runtime performance**.

Following the optimisation of the backup restore, the backup process was improved to reduce porting difficulties by replacing the **native** library. I re-implemented RSync as a pure Java implementation. Careful handling of overflows is needed in implementing the **rolling checksum**. In order to achieve good performance, **heap sort** was implemented on signature indexing so as to further reduce memory overhead.

Implemented a Microsoft Windows kernel filter driver that continuously records which blocks changes per file. The list of changed blocks is stored in a compact bitmap representation. Internally it uses Two-Level Segregated Fit (**TLSF**) **memory allocator** to manage data structure memory allocations. The bitmaps of block changes are flushed from kernel space into userspace for use by the Java based backup logic to narrow down the portions of files that have changed and need to be backed up.

Implemented various **Java** wrapper libraries providing integration into native Microsoft Windows components. These included: the **NTFS Change Journal**, the **VSS** (volume shadow copy service), **Knownfolders**, **MAPI** e-mail change monitoring. Integration was achieved via the use of **J/Invoke** and **JNA**. Many of integration libraries need to correctly handle 32-bit and 64-bit versions.

Developed an Audit Tool and Deployment Tool. These were implemented in 'C' and used **cURL** to provide HTTPS communications using **JSON** payloads and self signed SSL certificates. The executables where launched, via **PsExec**, on LAN based machines and communicate their results back to a controlling server. The audit tool scanned the machines for disk usage, directory structures, network adapters, OS version numbers etc. The deployment tool enabled remote installation, of backup software, by first pulling down a larger installation image.

3. Discovery: *various projects from 2009-2012*

   Discovery switched over to processing pharmaceutical claims using ProPBM. As part of this, it acquired KO PBM for which I had provided the **Java SEDA** based systems for integration. These systems process about 10000 **medial aid** pharmaceutical claims an hour during peak times.

   These integration components where leveraged to facilitate integration into the Discovery medical aid claims processing systems. Following this, I developed number of ancillary integration components so as to automate the synchronisation between upstream systems in Discovery and the pharmaceutical system, ProPBM.

   Legacy medical aid systems involving web services and relational databases where also worked on.

4. Liberty Life: *February 2012*

   Solve problems in integration between **MQ Series** with Oracle PL/SQL when moving to a **64-bit** platform. This required reworking `mqpack.sql` to support `@mqput` and `@mqget`. In order to achieve this workarounds where introduced for transferring 64-bit pointers between PL/SQL and 'C'.

5. Synaq: *August-October 2011*

   Consulting regarding development life-cycle process, and systems architecture. This enabled the Synaq to continue more effectively with their own internal development.

6. ITTI: *November 2011*

   Developed a small proof of concept web application using Cohesion as the backend framework with JSON messaging to a JavaScript/HTML frontend. This was to provide simple data storage and retrieval for a local sheriff's office.

7. Emerging Markets Data Ventures: *September 2010*

   Consulting input and project guidance on systems architecture and software development for one of the clients internal projects.

8. KO, KO PBM, KOH, Bathabile, Agility: *various projects from 2006-2008*

   Developed various middleware integration components. These have all been developed in Java J2SE. These systems are modular **multithreaded** components designed for high throughput and heavy load by following the **SEDA** (Staged Event-Driven Architecture). Persistence is managed via **JDBC** and communications are managed via TCP/IP socket channels.
   These components enable the integration of medical claim processing systems with custom medial administration systems.

9. Nettrace: *December 2007 to October 2008, 2010*

   Systems development in **C# .Net** of management software for computer asset tracking processes. This includes multithreaded components, relational database access and HTTP network communication. A minor enhancement was later made to support GZIP compression.

10. VWV: *August 2008*

    Developed a low level integration solution enabling communication between audio-visual hardware using serial **RS232**.

11. Rennies Travel: *September 2006 to October 2007*

    Consulting services provided regarding technology directions and new development being undertaken by Rennies Travel. The reports covered business process analysis as well as reviews and guidelines on software development process.

12. OPSI: *July 2007*

    Presented a short workshop on **unit testing** and agile software development.

13. Cyest: *February 2004 to October 2006*

    Acted as architect and senior developer for a Sales Performance Management System. This business application was developed on top of the J2EE framework. This application uses Hibernate for persistence. It leverages XML and browser support for grids to present a low bandwidth web application with a responsive frontend.

14. Knowledge Objects: *December 2004 to April 2005, 2010*

    Architected, designed and built a client-server middleware layer used to facilitate the integration of Knowledge Objects' Knowledge Power system with the applications used or developed by their clients.
    The client side is written in **C** and the server is a multithreaded server developed in **C++**. The development concentrates on cross-platform support and robustness, and was later ported from HPUX to Linux.

15. Magnolia: *June 2004 to December 2004*

    Architected, designed and built a Tomcat hosted web application. This presented a navigation frontend for retrieving multimedia content with the navigation structure being dynamically built from associated metadata.

4.3 Striata

February 2003 to February 2005

Striata's main business is bulk e-mailing. It specifically targets sending encrypted e-mail such as monthly statements. In my position as **senior developer** I designed and developed various components that support and comprise the backend infrastructure and provide developer tools. Projects included:

General: I was a member of the Architecture and Design steering committee. This included an regularly reviewing the overall architecture of the systems as well as being party to the design meetings for new projects or providing feedback to other developers.

I also provided mentoring and general help to other members of the team.

Component Versioning: This tool is used entirely by the developers. It provides a scalable way of tracking individual components, the dependencies in between components and the high level changes made to them. The result is a comprehensive view of the software components, change logs, automatically incremented version numbers and a top level build number. This is useful for tracking development progress and isolating bugs.

Hardware Security Module Integration: nCipher produce hardware security modules. I integrated this with Striata's mailing engine so that it could be used to send signed e-mails. I performed a similar integration with the Thales Websentry device. To achieve this I worked with OpenSSL Engines and the PKCS11 API.

CPPSubstTemplate: To perform bulk e-mailing, templates are created and substitutions are performed for each outgoing message. This project redeveloped one of the core **Perl** modules by providing a **C++** library that managed storage and manipulation of the data that is to be substituted into the template. This library is interfaced with Perl via the **Perl XS** layer. The benefits include better portability between C++ and Perl (as well as potential for other languages), better data **serialisation** capabilities and more options for future improvements.

Document Repository: This backend **server** process manages storage and **indexing** of arbitrary documents. The documents are stored as BLOBs and custom indexers are provided to extract metadata from different document types (e.g. plain text, CSV, SubstTemplate data, MS Word etc.). Schemata are configured to map index data into extracted metadata. The metadata can be searched using a simple **query language**. The result is a document store where documents can be retrieved by **searching** the **metadata**.

FOP Server: In providing bulk e-mail facilities it is often advantageous to be able to generate PDF documents. **FOP** is a specification that describes how to transform XML and XSLT into a PDF document. This server **integrates** the **Apache Java FOP** implementation with **C++** to provide an efficient stand-alone robust system that can be accessed via **RPC** to generate PDFs.

Meta Data Engine: Document Repository was used as the basis for then building an independent library that focused on handling the metadata storage, indexing and searching. This also integrates with the

query language. This component provides a clean interface whilst making it easier to develop optimised routines.

RPC Encryption: Striata uses a proprietary cross-platform **Remote Procedure Call** (RPC) mechanism. I was tasked with extending the communication library to support **encryption** and **cryptographic authentication**. To do this I first implemented **Diffie-Hellman** and the **Digital Signature Algorithm** (DSA) in a **cross-platform** independent library and then used this to build a secure **C++ streambuf**. This was then slotted into the current RPC implementation.

Search and Resend: In this project a **web-frontend** was built using **Perl** to provide Storage, Search and Resend functionality for bank statements. The system was built on top of the Document Repository and provided facilities for viewing and resending statements via e-mail or SMS.

Thin Consolidation: This system provided a central hub that enables a presenter to provide a consolidated view of bills to a customer. The presenter would usually be a bank and can display bills and provide a method of payment. The biller may be any company that offers services that will need to be billed. The system uses a **SOAP** like communication layer for interfacing with presenters.

Widget Framework: I built a full **MVC** widget framework for developing web applications in Perl. This framework splits the web app into separate widgets that are implemented as model (state), view (render to user), controller (collect actions). Widgets implement the **Observer pattern** by allowing listeners to attach and be given notifications of changes.

4.4 Part-Time (during MSc)

1. Liberty Life: *Independent Consulting April 2000 to March 2002*

   Net Query: Liberty uses a proprietary single-sign-on solution. This system used Sybase SNI for communications. SNI is no-longer a supported communication mechanism. I was contracted to build a replacement that would support the authentication system that handles over one million transactions per day.

   The resultant system was developed in **C++** and same code base supports **IBM OpenEdition MVS** UNIX, **Sun Solaris** and **Linux**. On OMVS, COBOL code is used to integrate with legacy backends. The system is a **multithreaded** and **multiprocess client/server** system that uses TCP/IP for network communication. The transport subsystem has the following layers: data marshaling, stream compression, packet construction and encryption.

   MMS: I provided information and advice regarding their proprietary messaging system (MMS).

   NetServTester: I built a network service monitoring tool for monitoring servers that use text based protocols such as HTTP, SMTP and POP. The status was decided using send/expect request and response pairs. This tool could be deployed under **Microsoft Windows NT** as a **service** or on UNIX as a **daemon**. It was integrated with exiting tools and provided HTML status reports and extensive logging.

   Server Security: I was contracted to provide an assessment of the **network** and **host security** of certain servers as well as provide recommendations for improving the security. This included standard best-practises and user education.

   Web Server Authentication: I performed maintenance and enhancements to the authentication infrastructure used for the web servers.

2. DataServ: *Contracts in between July 2001 and November 2001*

Dial-on-demand: I built a dial-on-demand solution for use in small office parks. The system worked over an ISDN connection. The server performs IP-masquerading and also acted as an e-mail gateway for SMTP and POP as well as working as a caching DNS.

3. Multitrade: *Small contracts in between January 2000 and March 2001*

   Linux installation: Multitrade provides on-line commerce facilities. The systems run using the **Linux operating system** and I was employed to **install** and **configure** Linux, as well as provide support regarding the **administration** of the server.

4. Xyan: *Independent Consulting April 2000 to December 2000*

   Venue Directory: As part of a team of three I helped develop the complete **web application** which provides the primary functionality for `http://www.venuedirectory.com`. This site enables customers to search for hotels/venue around the world and find out more about their facilities. The system was implemented using Open Source components (e.g. Linux, Apache/Cocoon, Tomcat and PostgreSQL).

5. Wasp/Cellpoint: *Independent Consulting September 1999 to November 1999*

   SimCell: I was contracted to developed a **Java** web **applet** that simulated a Cellular Mobile Phone for Wasp/Cellpoint. The applet was to be used to demonstrate the capabilities of the smart card applications being developed by Wasp/Cellpoint. The applet was also integrated with Wasp/Cellpoint's SMS infrastructure so that the user could send an **SMS** to a real GSM phone.

6. OPSI: *Independent Consulting March 1999 to September 1999*

   Order Management: I was contracted to rework an existing Order Management and Stock Control System that was developed by OPSI for a stationery factory. The system was developed using Microsoft Access and I developed new screens, reports and database layouts. I also integrated the system with MAPI to provide e-mail facilities.

4.5 Liberty Life

February 1997 to January 1999

After receiving a bursary from Liberty Life I was employed in their IT Projects Department.

Authentication Framework: I built a system that provided a uniform mechanism authenticating users accessing web servers on the Liberty network. The core component was a central **authentication server** that supported multiple authentication schemes. The primary authentication mechanism was a legacy mechanism that involved passing requests via Sybase SNI to the mainframe where RACF authenticated the username and password. Client modules where then built for Apache, Netscape and Microsoft IIS web servers. The authentication services where also integrated into other systems.

Intranet/Web Sites: I helped with the deployment of the internal intranet. This included web server architecture, HTML web page design and development of CGI applications.

Log Server: UDP base client/server framework for logging trace messages generated by applications.

MQ Series Framework: I was the **team leader** and mentor of a team that was tasked to develop a generic API and **middleware** infrastructure to facilitate the implementations of **MQ Series** at Liberty Life. The development natively supported Microsoft Windows NT and Sun Solaris.

Messaging Engine: Liberty services its insurance brokers and agents in the field via a proprietary multi-layered messaging system (MMS). This provides message transport from end-users desktops to the mainframe. I performed extensive maintenance to improve robustness. I also developed extensions for the system.

4.6 Part-Time (during undergrad and BSc)

1. NeuralWare: Apache Module *Part-time May 1996 to December 1996*

   Apache Module: I built an **Apache module** that formed part of an Advert Server being developed by NeuralWare. The component performed substitutions on special tags in standard HTML documents.

5 Skills Matrix

Skills Matrix

Level Key: Novice, Basic, Intermediate, Advanced, Expert.

Skill	Duration	Last Used	Level
Actor Model	2 years	2014	Advanced
The actor model is a particular pattern for implementing highly concurrent systems while helping to ensure thread safety by ensuring that any given object (actor) can only process one message at a time. The actor model is often used in conjunction with event driven systems.			
Related Projects: Gethos,Nashua Communications			
Ant	10 years	2014	Advanced
Created build systems for individual projects as well as Ant based build system templates for reuse by development teams.			
Related Projects: Gethos,Nashua Communications			
Apache Development	1 year	1998	Intermediate
Server modules developed in 'C'. This includes providing authentication hooks as well as content filtering.			
Related Projects: Authentication Framework			
Architecture & Design	10 years	2014	Advanced
Architecture and design skills and patterns are key for developing and maintaining cleanly implemented and decomposed systems. This applies at any scale, from simple programs to large distributed systems.			
Related Projects: Gethos,Synaq,Discovery,Cibecs,Nashua Communications			
C	6 years	2013	Expert
I have developed using plain 'C'. This is good for integration with other languages (e.g. binding to Perl, Java or Visual Basic) and for low level development such as Operating Systems or building higher level languages. Concepts include memory management, macros, pointers, function pointers.			
Related Projects: FOP Server, CPPSubstTemplate, Messaging Engine, Authentication Framework			
C++	7 years	2010	Expert
In addition to pure 'C', I have developed complex systems using C++ that leverages the object-oriented support of the language and related aspects such as classes, abstract classes, objects, inheritance, multiple inheritance, templates, operator overloading, exception handling, virtual functions, virtual destructors, polymorphism, STL, macros, name spaces. The applications developed include multithreaded applications and network services. I primarily use gcc (under Linux, Solaris, Cygwin, FreeBSD) and Microsoft Visual C++ compilers but I have also used the Sun compiler for Solaris, the HPUX compiler, and the IBM compiler for OMVS.			
Related Projects: Meta Data Engine, CPPSubstTemplate, Document Repository, Net Query, NetServTester, MQ Series Framework,			
C#	2 years	2010	Intermediate
C# development of back-end components together with NUnit for testing.			
Related Projects: Nettrace			
CAS	2 years	2013	Intermediate
Compare-And Swap constructs can be used to implement lock free concurrent algorithms.			
Related Projects: GethosCohesion			
Cryptography	1 year	2008	Advanced
In have been awarded an MSc in Mathematics for my thesis in Cryptography. As part of my MSc I implemented DES in C++. Additionally in a commercial setting I have implemented the Diffie-Hellman and DSA algorithms in C++. I have also used block ciphers such as Blowfish, DES and AES.			
Related Projects: nCipher Integration, RPC Encryption, Net Query, Mathematics MSc			

Debugging	10 years	2014	Advanced
Various debugging techniques and tools e.g. jvisualvm, gdb, strace, jmap, ldd			
Git,SVN,CVS	15 years	2014	Advanced
Version control is vitally important for the management of software projects. Over time the de-facto standard has changed and I have used CVS,SVN,Git. These have been used for branch management, tagging, patch merging etc.			
Related Projects: Striata Projects, Liberty Life Projects			
Event-Driven Systems	2 years	2013	Intermediate
Developed an actor based framework for building event driven systems.			
Related Projects:			
HTML	10 years	2014	Advanced
I have developed numerous sites using HTML. This include basic layout, forms, layers, JavaScript.			
Related Projects: Search and Resend, Venue Directory, Intranet/Web Sites, Gethos, Discovery, Nashua Communications			
IPC	10 months	2002	Advanced
Inter-Process Communication is needed in applications that are distributed over multiple processes. This model pre-dates multithreaded code and is still appropriate in some situations (e.g. enhance security be separating tasks, robustness, legacy systems that do not support multithreaded environments). Concepts include semaphores, message queues and shared memory.			
Related Projects: Net Query			
Ivy	3 years	2013	Advanced
Ivy provides dependency management as part of Ant builds.			
Related Projects: Gethos, Nashua Communications			
J2EE	1 year	2009	Intermediate
I have built applications on top of Tomcat as well as J2EE applications using JBoss and Websphere. Combined with this I have used Hibernate for persistence.			
Related Projects: MeDiA,SPMS			
Java	10 years	2014	Unknown
Java is a good object-oriented modern language that provides platform (hardware and operating system) independence. I have developed web applications, backend servers and user-interface applets using Java. This includes concepts such as classes, inheritance, interfaces, exceptions, collections, swing.			
Related Projects: FOP Server, SimCell, Venue Directory, Cibecs, Nashua Communications, Gethos			
JDBC	4 year	2012	Intermediate
Java Database Connectivity (JDBC) provides access to relational databases from Java with queries and updates being written in SQL.			
Related Projects: Discovery			
JNA	1 year	2012	Advanced
Java Native Access (JNA) provides access to native C shared libraries via a pure Java interface.			
Related Projects: Cibecs			
JNI	2 months	2004	Intermediate
JNI is a Java Virtual Machine API that enables Java code to interface with 'C' routines. I used it to embed a Java sub-system into a C++ application that used Java code to implement the bulk of the logic of the system where as the C++ layer was used to manage integration with other applications.			
Related Projects: FOP Server			
JUnit	9 years	2014	Advanced
JUnit provides a framework for development repeatable unit tests when coding in Java.			
Related Projects: Gethos,Cibecs,Nashua Communications			
JSON	2 years	2013	Advanced

JavaScript Object Notation (JSON) has gained favour as a text representation for messages. This format is useful for backend to web frontend communications, or in RESTful transfers of state.			
Related Projects: Cibecs, Gethos, Nashua Communications			
Linux	10 years	2014	Expert
This includes admin, user application development and kernel development as well as desktop usage.			
make	6 years	2010	Expert
Most project need a build system. `make` is the de-facto standard for building C/C++ applications under UNIX. It can also be used for building under Windows. I use it to manage dependencies and drive compiling. Functions and conditionals can be used to support more complicated tasks or to produce a cleaner more usable build environment.			
Related Projects: Striata Projects, Net Query			
Mathematics	N/A	2002	Advanced
MSc including cryptography, number theory, analysis of algorithms and stochastic calculus.			
Middleware	6 years	2002	Advanced
Many of my projects have been involved in integrating distributed systems. For this middleware communication layers have been needed. This also enables reuse of legacy systems via more modern frontends.			
Related Projects: Net Query, MQ Series Framework, Messaging Engine			
MQ Series	8 months	1999	Intermediate
I constructed a high level framework that included queue management and trigger monitors in a cross-platform manner.			
Related Projects: MQ Series Framework			
Multi-platform programming	4 years	2004	Advanced
Simultaneous support for combinations of Solaris, Linux, MS Windows NT, IBM OpenEdition MVS UNIX			
Related Projects: Document Repository, Net Query, NetServTester, MQ Series Framework			
Multi-threaded programming	10 years	2014	Expert
Many of the applications I have developed are server-side and support requests from multiple clients. To enable this, these applications need to allow for multiple threads of execution. The tools used to support this include mutexes, spin-locks, barriers, conditional-variables, thread-local-storage, compare-and-swap. I have worked with the POSIX threading library as well as proprietary threading libraries and Java threading. Patterns that aid development of multithreaded applications are lock-pointers, guard objects and producer-consumer queues.			
Related Projects: FOP Server, Document Repository, Net Query, Gethos, Discovery, Nashua Communications			
Non-Block I/O	4 years	2013	Advanced
I have implemented non-block I/O based multiplexors for the purpose of scalable network traffic handling in Java.			
Related Projects: Cohesion, Nashua Communications			
Network programming	8 years	2013	Advanced
I have performed a range programming tasks. This includes protocol design, TCP/IP and UDP/IP applications, building client and server applications, socket programming and data marshaling.			
Related Projects: RPC Encryption, Authentication Framework			
ODBC	6 months	2003	Intermediate
In developing portable backend servers that use a database I used ODBC. Under UNIX I used iODBC as the driver manager and under Windows the default ODBC driver manager. ODBC was used to connect to PostgreSQL, IBM DB2, mySQL and Microsoft SQL.			
Related Projects: Document Repository			
OOD/OOP	12 years	2013	Expert

Object-Oriented Design and Programming is one of the most important methodologies in use currently. In implementing an OO Design I find that it is important to understand how the chosen programming language supports and facilitates the development. The general concepts include classes, inheritance and data encapsulation. There are also many other design patterns that aid OOD.			
Related Projects: Striata Projects			
Parsing	2 months	2003	Advanced
I used a parser generator for C++ (Boost Spirit) to parse a special purpose query language that was then translated into SQL queries.			
Related Projects: Document Repository			
Perl	1 year	2004	Advanced
I have developed programs that range from small utility scripts to large object-oriented critical systems. In this development I worked with packages, blessed objects, references, function references and regular expressions.			
Related Projects: Component Versioning, Thin Consolidation, CPPSubstTemplate			
Perl-XS	5 months	2004	Intermediate
XS provides an interface layer that enables Perl to invoke 'C' routines.			
Related Projects: CPPSubstTemplate			
RPC	2 year	2004	Intermediate
I have developed using standards based and proprietary RPC frameworks. I also extended and performed maintenance on the same proprietary RPC framework. This includes working with concepts such as client/server, IDL, generated stubs and implementation classes.			
Related Projects: Document Repository			
Security	1 year	2001	Advanced
Host and network services. This includes performing audits, encouraging best practises and deploying specific technology such as IP filtering or encryption.			
Related Projects: nCipher Integration, RPC Encryption, Server Security, Dial-on-demand			
Shell scripting	4 years	2004	Advanced
Shell scripting for UNIX system is important for quick admin tasks and also for software deployment.			
Related Projects: Dial-on-demand			
StatsD	6 months	2013	Intermediate
Implement UDP version of StatsD for Java to enable pervasive monitoring of distributed systems via aggregated metrics.			
Related Projects: Nashua Communications			
Smalltalk	N/A	2004	Novice
Non-commercial: I am interested in the Smalltalk programming language and have built simple applications for the purposes of learning. I have worked with Cincom VisualWorks and Squeak.			
SQL	3 year	2013	Intermediate
Many projects have used relational databases in the backend. I have worked with PostgreSQL, IBM DB2, MS Access, and mySQL. Includes server installation, query writing, table design, relational integrity and indexes.			
Related Projects: Document Repository, Meta Data Engine, Order Management, Discovery, Nashua Communications			
TCP/IP programming	6 years	2013	Advanced
I have developed clients and servers on various platforms that make use of TCP/IP communication.			
Related Projects: RPC Encryption, Net Query, NetServTester, Authentication Framework			
UDP/IP programming	2 year	2013	Basic
UDP based client/server communication.			
Related Projects: Log Server			
Unit Tests	10 years	2014	Expert

I endeavour to develop using a test first coding approach. This ensures that I have tests to exercise all code that comprises the system under development. This results is more robust systems with better architectures and more reusable components. Furthermore, there is less risk involved in refactoring and evolving the system for future needs.			
Related Projects: FOP Server, Component Versioning, Meta Data Engine,Gethos,Cibecs,Nashua Communications			
UNIX	5 years	2013	Advanced
Development, administration and installation. Includes Linux, HP-UX, IBM OMVS and Solaris.			
Visual Basic	6 months	1999	Intermediate
I extended the backed functionality and GUI frontend of a system using VB under Microsoft Access.			
Related Projects: Order Management			
VoIP and SIP	1 year	2014	Intermediate
Voice over IP (VoIP) and the Session Initiation Protocol (SIP) can be used to provide telephony over the Internet.			
Related Projects: Nashua Communications			
Web applications	10 years	2014	Advanced
I have developed web application over a range of technologies as the tool sets have changed. Earlier applications where CGI based and developed in C++ and Perl. Later systems used tools such as Apache Cocoon, Tomcat, J2EE. Finally moving on to the likes of GWT or pure HTML5/Javascript with libraries such as JQuery and D3. Earlier systems relied on form posts, while later systems rely on asynchronous messaging via XMLHttpRequest.			
Related Projects: Search and Resend, Venue Directory, Rennies Travel, Nashua Communications, Discovery			
Web server admin	3 years	2013	Intermediate
Includes Apache and Nginx installation and configuring virtual hosts, proxy servers, transparent proxies, user authentication.			
XML/XSLT	1 year	2010	Intermediate
I have used the XML mark-up language to provides a text based data description that is easy for machines and humans to read. I have used it in conjunction with XSLT and Apache Cocoon as the primary mechanism for rendering pages in a web application. I have also used XML to provide SOAP like messaging.			
Related Projects: Thin Consolidation, Venue Directory, Discovery			

6 References

Cibecs	Neal Dewing	+27 83 383 7462	<neald [at] cibecs.com>
Discovery	Karien Greef	+27 83 288 8349	
	Nathan Edelman	+27 78 801 1942	<nathane [at] discovery.co.za>
Knowledge Objects	David George	+27 82 565 3877	<david [at] knowledgeobjects.com>
KO PBM	Gerrit Moolman	+27 83 321 6303	<gerritm [at] discovery.co.za>
Nashua Communications	Brynn Andrew	+27 79 500 2679	<brynn.andrew [at] nashua-communications.com>
NeuralWare	Nir Oren	+44 79 8629 0494	<noren [at] csd.abdn.ac.uk>
OPSI	David Clark	+27 83 293 4020	<david.clark [at] opsi.co.za>
Rennies Travel	Antony Soicher	+27 83 556 4666	<as [at] neuralogic.net>
Striata	Nic Ramage	+27 11 503 6900	<nicr [at] striata.com>