

Eric C. Hill
Software Engineer/Generalist

575 Oxford Ave.
Palo Alto, CA 94306

eric@stochastic.com

Prefer initial contact by email

Summary: I've been programming for a bit more than 30 years and still enjoy it. I prefer to use object oriented methods and tools to get things done, but am comfortable getting my hands dirty in the bits when needed. I enjoy opportunities to explore new problem domains. While many of the users of my work have in turn been programmers, I'm pretty good at talking with specialists and non-specialists alike in a variety of fields outside of computation.

Work Experience:

Principal Software Engineer/Senior Architect, Plusmo, Inc., Santa Clara, CA July 2008-July 2009

Led development of REST based web site and services framework with server-side Python and client-side JavaScript, with content provided by web designers. Built object abstraction in Python for MySQL database as part of a larger library framework. Introduced functional style AJAX framework for JavaScript clients as well as other abstractions from more advanced platforms. Worked on several mobile phone based applications, both server and client side. Worked with WebKit internals to create server-side rendering engine.

Principal Software Engineer, PayPal, Inc., San Jose, CA

Sep. 2006-Oct. 2007

On the infrastructure team of the Core Technologies group, I was part of the effort to refactor a very large monolithic code base. The particular function of my team was to develop components to facilitate that effort. Components included a publisher/subscriber based message passing system and a client/server remote procedure call framework. My work included specification and mock-up of web-based administration tools for the message passing system, status monitoring for the RPC framework, and various tools to assist other developers using the RPC framework. The PayPal code base is mostly C++, with many development scripts in a variety of languages. Most of my scripting work was in Python, although some Perl work was called for. The administration tools involved creating dynamic HTML pages with a small amount of JavaScript.

Personal Time Out

Oct. 2001-Aug. 2006

While attending to personal matters, I stayed a bit busy helping friends with their software, networking and computer repair problems, as well as working on various software projects of my own, mostly in C++ and Java. I maintain a home network with several PCs and a Linux machine or two, using VPN technologies when traveling. I've made a point of continuing my studies in math, computer science and physics, among other things.

Senior Software Engineer, Agorics, Inc., Los Altos, CA

Jan. 1994-Oct. 2001

As a founder and software engineer at Agorics, I developed software in both product and research environments for a variety of clients. Agorics' company mission was to develop software platform technology, principally for secure, distributed systems, especially in the area of online economies. Customer funded, Agorics did contract work in a variety of areas.

My first assignment was the development of a Smalltalk API for a commercial C++ oriented object database. This supported persistence for arbitrary Smalltalk objects as well as mapping of database structures into Smalltalk with full database functionality. Features included garbage collection of database

structures and type migration when Smalltalk types were changed. The API was included in the client's product line.

Later I was assigned to Sun Microsystems Laboratories where I worked on the design and development of a persistent secure distributed object system in C++. This was designed with Internet commerce in mind and for securing corporate and intercorporate communications and related functions. Applications included various payment systems and online auctions, bandwidth allocation, online barter, and content delivery. Payment systems included micropayments, stored value cards over TCP/IP, and basic dollars and cents for product or service delivered. A primary concern of Agorics was software safety, both in the traditional sense of security and in the sense of reliability; this included the ability to securely exchange executable content. When Java emerged, we continued to develop communications and commerce related tools, this time using RMI for object distribution, but using our own persistence system.

I also worked on the FSTC eCheck project where Agorics initially worked with Sun Microsystems and later with IBM. We provided much of the implementation and design effort for IBM's eCheck deposit server which was deployed at Bank Boston and at NationsBank. Many other vendors (for example, BBN) and institutions (e.g., Boston Federal Reserve Bank) were involved and we had direct dealings with most of them. The server received eChecks in the form of formatted email, performed PKCS verification and decryption. It also verified the validity of the deposit and sent it to the bank's backend. I was responsible for several of its components, including the top level server logic.

I was sent to IBM's T.J. Watson Labs in Hawthorne, NY for six weeks to integrate Visa's SET technology into Sun's Java Wallet.

Between outside contracts I worked on various Agorics internal projects such as:

- A secure, role-based corporate email server
- A locally developed secure distributed object oriented programming language
- Uses for KeyKOS, a capability security based microkernel for which Agorics had obtained rights.

Member of Technical Staff, Memex, Inc., Palo Alto, CA

Jan. 1993-June 1993

At Memex I continued to work on the Xanadu software development project after the technology was acquired by the recently founded Memex. My job description was essentially the same as at Xanadu until development discontinued for lack of funding.

Member of Technical Staff, Xanadu Operating Co., Palo Alto, CA

May 1988-Jan. 1993

Xanadu was an early hypertext system. Its founder, Ted Nelson, coined the word, "hypertext," as well as many other terms in common use today. I was an early project member, with many roles during my tenure there.

I was the lead software integrator for a team of about 10 programmers. I co-developed and maintained our core library in Smalltalk and C++. The principle requirement for the C++ library was to provide a runtime environment for code automatically translated from Smalltalk. Storage management with garbage collection, finalization support, and primitive collection types were among the key requirements. One of my main contributions here was a non-conservative generational garbage collector for a large virtual memory with post-mortem finalization, based on the works of Baker, Ungar, and Deutsch. Smalltalk work included development of data structures and other support classes for the hypertext engine as well as work on the underlying development environment. I developed debugging and performance analysis tools for C++, using some SPARC assembly language and binary file features. Work in both Smalltalk and C++ included repair, analysis and improvement of performance of code written by other team members. I performed maintenance and customization of build tools as well as some system and network administration.

Languages, etc.

C, C++, HTML, Java, JavaScript, AJAX/JSON, MySQL, Python, Smalltalk, XML, Various Assemblers, others

Platforms MS Windows, Cygwin, Linux, Varieties of UNIX, MS-DOS, others

Education

Bachelor of Science in Mathematics with C.S. Emphasis, University of Arizona at Tucson, AZ May 1988

(Physics through 4th semester)

Avocations

Reading. Math, Science, Technology, Language, History, Philosophy, the occasional classic, etc.

Recreational programming: simulations, fractals, mathematics, art, exploring new programming ideas, etc. Beach combing. Car trips. Photography. Gardening. Helping friends with their computers.

References available on request