
Précis

Veteran Unix and Macintosh generalist with strong domain knowledge and extensive work experience in digital video, relational databases, and large, distributed systems. And Emacs.

<i>Languages</i>	Objective-C, Ruby, C, Python, elisp, Java
<i>API</i>	QuickTime, ffmpeg, Cocoa/Foundation, POSIX
<i>Tools</i>	Emacs, XCode, Unix shell, Subversion, git, CVS
<i>Databases</i>	Postgres, Redis, Oracle

Current Position

- 2011 – Present **Video Software Engineer**, *Zencoder, Inc*, San Francisco, CA.
Video processing software development for a successful Y-Combinator funded cloud-based video encoding startup.

Work Experience

- 2009 – 2011 **Co-founder**, *Pipely Inc.*, San Francisco, CA.
Engineering and product development for a Y-Combinator funded social video application. After rapidly iterating over several ideas (stabilizing and color-correcting user-submitted mobile phone video; delivering YouTube content via adaptive HTTP streaming to Apple iOS devices) we settled on Nowmov, a social discovery and personalization system for online video.
- Managing the engineering of a small consumer Web start-up, including scheduling, task assignment and monitoring, and setting engineering standards and directions;
 - wrote an automagic caching system to sit between Rails and Postgres, with the Redis K/V store and the Sequel ORM layer;
 - built an email-to-upload video processing system, in Rails, shell, and ffmpeg;
 - miscellaneous founder's tasks – fund-raising; shopping for office space; interviewing, hiring and firing; *being* interviewed; paying for coffee.
- 2004 – 2009 **Principal Video Software Engineer**, *Apple, Inc.*, Cupertino, CA.
Responsible for the design and construction of the iTunes Store video ingestion system, from defining partner and output file formats to writing the QuickTime-based transcoding software, to researching and implementing novel methods for automated video quality control.
- Developed tools for transcoding, scaling, and color-correcting video to be played on a wide variety of hardware devices;
 - wrote a novel, video-specific queuing system for distributing tasks across a large cluster of unreliable machines;
 - created a context-aware system for encoding H.264 in parallel;
 - designed and implemented automatic fault detection, including correlating audio energy levels with luma information and perceptual hashing of a reference encode;
 - designed and implemented a mechanism for capturing and preserving transient data from the H.264 encoding process, previously lost, by leveraging the QuickTime API;
 - served as the Store's subject matter expert in meetings with partners, vendors, and other teams within Apple.

2002 – 2004 **Senior Database Engineer**, *iParadigms, LLC*, Oakland, CA.

Designed, built, and maintained a very high performance Postgres cluster for Turnitin.com, the world's largest automated anti-plagiarism service.

Designed, implemented, and supported an eight-node single-writer/many-reader clustered database, with hot spares and a near-real time backup;

wrote a custom query load-balancer, implemented as a library for both Perl and Python applications, that dynamically routed queries to the most appropriate database instance;

served as chief database architect, working with other engineers on designing database schemata and optimizing query performance;

administered development resources, including Subversion and Bugzilla instances.

2001 – 2002 **Contract Engineer**, *Cisco Systems*, San Jose, CA.

Wrote a custom application-layer proxy to allow for legacy Perl web applications (developed in a mixture of mod_perl and CGI) to be integrated into a pure Java/J2EE web framework;

administered the development team's Solaris server, including maintaining CVS and Apache services.

1999 – 2000 **Contract Engineer**, *Qwest*, Minneapolis, MN.

Web application development, in Java, Python, and Perl;

designed and built an integration framework, based on XML-RPC, for legacy Smalltalk applications to make use of new, web-based services.

1998 – 1999 **Contract Engineer**, *Digital River, Inc.*, Eden Prairie, MN.

Built the server side to the Digital River upgrade management and download tool; over 20,000 lines of object-oriented Perl, implementing a custom wire protocol; a high-performance cursor caching proxy for the DBI; flexible syslog and console logging; robust signal handling and runtime module loading;

wrote a restartable, multi-threaded FTP-based file distribution mechanism for Digital River's partners to facilitate batch order processing;

designed and built a secure, asynchronous order processing and fulfillment system based on INN and NNTP, written in Perl and C.

1997 – 1998 **Contract Engineer**, *Genesys, LLC*, St. Paul, MN.

Wrote a PL/SQL regression testing tool, in Perl, including a (nearly-) complete PL/SQL parser;

extending, fixing and upgrading features in the company's flagship product, a workflow management system for the prepress catalog and newspaper supplement industry, mostly in Perl, Javascript, and PL/SQL;

Wrote a Quark Xpress plugin for the Macintosh that integrated with the company's flagship workflow management product.

1993 – 1997 **Systems Programmer**, *The University Of Chicago*, Chicago, IL.

General Unix systems programming and administration in an academic environment

Designed, built, and managed the Center for Anthropological Computing, a unique multimedia data-collection and analysis facility; including delivering cross-platform (MPEG-1) video over IP; a simple object database (custom built); video, audio, and still image acquisition, indexing, and storage;

C, Python and Perl programming in support of 10,000 shell users of the University's central computing systems (SunOS, IRIX, HP-UX, Linux);

wrote a custom load-balancing library and accompanying tools, in C and Perl, for managing long-running statistical software jobs (running in SPSS, SAS, Matlab) on a heterogenous Unix cluster (IRIX, SunOS/Solaris, Digital Unix, AIX, Linux, OSF/1).

Patents

61/410785 A METHOD AND SYSTEM FOR CONTENT DISCOVERY THROUGH SOCIAL NETWORKS (PROVISIONAL)
20080065691 METADATA FOR PROVIDING MEDIA CONTENT