

Damien Katz

Objective

I am looking for an engineering and leadership position within a quality software development organization. My primary interests and experience are in information management and collaborative technologies; tools that let people organize, view and share data.

My ideal employer will have challenging problems and be filled with bright, creative (and quirky) people who take pride in their work and want to change the world for the better..

Professional experience

2005 – 2007

CouchDb Project

Charlotte, NC

Architect

- CouchDb is an open source, document database system designed for document oriented and collaborative applications.
- Built on Erlang OTP, a concurrency-oriented functional programming language with an extreme emphasis on fault-tolerance and reliability.
- Uses a schema-less document data model, with a formula-based “view“ facility for filtering, formatting and collating the database contents dynamically into tables.
- CouchDb is a peer-based, distributed database system. It allows any number of replicas of a database to exist on remote machines, where each replica is a completely independent, queryable and updatable database. Changes in databases (additions, edits and deletions) can be replicated, one way or bi-directionally, to other replicas. CouchDb has built-in conflict detection and management.
- The core database engine and server is written in Erlang. The Fabric engine is written in C++ and Antlr (a compiler creation tool). CouchDb uses a built-in HTTP server and exposes an XML programmatic interface. The ICU library is integrated to provide full UTF8 and internationalization support.
- I’ve designed and coded CouchDb from scratch by myself. Including:
 - Optimistic update model and highly concurrent, lock-free server design. Update operations are serialized, queries and reads are executed concurrently and are never blocked by writes.
 - Crash-only database engine. Start-up consistency checks are never necessary. Non-destructive storage updates and append-only b+ trees ensure the database file is never in an inconsistent state.
 - Fabric, a query and validation language, designed specifically for the CouchDb schema-less document model and easy text manipulation (including Perl style regular expressions). It includes a language compiler, runtime engine and test suite. It interfaces with Erlang using separate OS processes and piped interfaces.
 - Efficient, incremental view indexing engine.
 - Full UTF8 text support: XML, Erlang view collation, the Fabric compiler and runtime engine.
 - Robust, incremental replication with built-in conflict detection and management.

- RESTful XML over HTTP programmatic interfaces.
- Integration. I've dealt with many bugs, undocumented behaviors, perplexing APIs, and quirks in the tools and technologies. Every problem, regardless how baffling, frustrating and time consuming that's come up, I've dealt with and made work right.
- Portable code. Builds and runs on: Win32, Linux x86, Linux AMD 64, OS X PPC, OS X x86, BSD x86.
- Project Home: <http://couchdb.com/>
Documentation: <http://couchdbwiki.com/>
Technical Overview:
http://www.couchdbwiki.com/index.php?title=Technical_Overview

2002 – 2004

Kubi Software

Lincoln, MA

Engineer

- Brought on to help complete Kubi, a secure distributed collaboration platform that integrated with MS Outlook and Lotus Notes. Remote clients replicated changes via specially encoded email messages that were extracted by other participating clients.
- Responsible for the Notes platform development and maintenance and some parts of the core platform. This was mostly back-end C++ development, but also included UI development.
- Designed and built a distributed testing framework to automate system and load testing. The tests and testing framework were written in Python, and used custom RPCs to remotely control C++ Kubi clients. This allowed automated multi-machine testing of the fully distributed systems, which previously was a manual process.
- Built various internal tools for debugging, building and analyzing the product in Python, C, C++ and VBScript.
- Kubi ultimately failed, its design was flawed in several ways (perhaps the biggest mistake was building the replication system on SMTP). But the startup experience taught me many technical, project and business lessons.

1997 - 2002

IBM/Iris Associates

Westford, MA

Engineer

- Lotus Notes and Domino project engineer and developer.
- Single-handedly redesigned and implemented the Notes Formula Language runtime engine in C++. Notes Formula Language is a simple language, similar to spreadsheet formulas, built into the core and used throughout both Notes and Domino. It is the primary programming language used in Notes applications and runs both client side and server side.

Critical for server performance, it is used for generating database views and indexes, and on the web server used to generate dynamic pages and validate submissions.

Details:

- Designed and implemented a completely new internal runtime execution model, while maintaining full backwards compatibility.
- Increased runtime performance 200-400%.
- Lazy evaluation allows expressions to evaluate only when needed.
- Partial result caching allows intermediate computation results to be

computed once and used again in later executions.

- Increased the Formula Language compiler performance by 400%.

- More details:

http://www-128.ibm.com/developerworks/lotus/library/ls-meet_dev_Damien_Katz/

<http://damienkatz.net/2005/01/formula-engine-rewrite.html>

- I helped the database team during the push to get 6.0 shipped.. Notes and Domino uses its own proprietary, cross platform database engine. I fixed dozens of bugs and performance problems.
- Responsible for maintaining unread mark functionality, both the front-end Notes client and back-end database and server level. Stabilized the code base fixing dozens of bugs and then designed an efficient and reliable system for replicating unread marks between servers.
- Implemented Java calling JavaScript Liveconnect functionality in the Notes client. Used message queues to allow multiple Java threads to transparently run code in the single threaded Notes client.
- Helped maintain the Netscape-based JavaScript engine that is integrated into the Notes client.
- Created a popular open-source LotusScript library, which allows programmatic access and modification to database design elements.
- Designed and implemented the R5 Welcome Page (the Notes client home page) as a customizable “dashboard”, giving users the ability to choose from different layouts and content sources.
- Developed the “Database Subscriptions” front end, which allow users to subscribe to a Notes Database and receive content matching custom criteria.
- Redesigned Statistics and Events, and Statistics Reporting administrative front end, for easier set up and maintenance for administrators.
- Developed features for Mail and Calendaring & Scheduling.
- A constant source of bad jokes and headaches for managers. (But in a good way!)

1996 - 1997

Unity Consulting

Charlotte, NC

President, CEO and Janitor

- Performed Lotus Notes consulting work in the Charlotte area.
- Developed an Engineering Approval Cycle database for a Fortune 500 corporation. Included user definable workflow allowing for multi-step complex cycles, and automatically created markup and change history documentation.
- Developed a prepackaged, commercial Sales Force Automation system for Lotus Notes 4.5. Sold 0 copies, but then I suck at selling. I sold the source code for \$1000. (Not a total failure!)

1995 - 1996

PRIMA Consulting

Charlotte, NC

Consultant

- Performed Lotus Notes consulting work in the North Carolina and South Carolina area, developing and extending custom applications and infrastructure.

Commercial products shipped

- Lotus Notes & Domino 5.0 and 6.0
- Kubi Client for Notes 1.0, 1.1, 1.2

Summary of qualifications

- Kubi Server for Domino 1.1
- Experience in engineering and development of large, mature commercial software packages in corporate environment, as well as dealing with rapidly evolving code and requirements in a start-up.
- I work well in teams, but I produce my best design and coding work alone.
- Proven design and problem solving skills. Designed and implemented a diverse range of working software components, systems and solutions on a variety of platforms.
- Proven coding skills and very detail- and correctness-oriented.
- Strong written communication skills. I've written specifications, project plans, project documentation, developer documentation, end user documentation, articles in trade journals and I maintain a development oriented blog (damienkatz.net).
- Excellent debugging skills. I tend to keep my bug count low. Most large projects I've worked on, I've spent most of the end of the project helping other developers and teams with their bugs queues.
- Strong UI and HCI design skills.
- Programming Languages: C++, C, Erlang, Java, LotusScript, JavaScript, Python, PHP, Antlr
- Various buzzwords: distributed computing, replication, OOP, Functional programming, XML, HTTP, HTML, REST, piped interfaces, concurrent programming (shared state threading and mutexes; process based asynchronous messaging), fault tolerance.
- Understand portability issues at a low level. I've written large bodies of high performance C/C++ code that simultaneously builds and runs cleanly on 7 different platforms (Win32, Linux, AIX, Solaris, Macintosh, AS/400, HP UX), dealing with pointer size, memory alignment and endian issues.

Education

1991 – 1995 North Carolina State University Raleigh, NC

Computer Science

Other

- Speaker at conferences in Orlando, Las Vegas, Berlin, Dusseldorf and London with audiences as large as 2,500 people.
- Patent applied for Notes & Domino unread mark replication.
- Contributing author *Lotus Notes and Domino 4.6 Unleashed*, Sam's Publishing, 1997
- Author *Customizing the Welcome Page for Your Corporation*. [Lotus Developer Domain](#)
- Author *Inside the Rnext Formula Language* [THE VIEW](#).
- Blog: <http://damienkatz.net> (25k+/monthly unique visitor IPs)
- I once battered and deep fried a Snickers bar. It was awesome. <http://damienkatz.net/2005/02/deep-fried-candy-bars-illustrated.html>
- I once battered and deep fried a Twizzler. It was disappointing.