Lucas Beeler

lsbeeler@acm.org ■ (650)395-7716 ■ http://www.lucasbeeler.net ■ Fremont, California 94538

overview

- Software professional with 6 years of experience in a broad mix of roles including senior software engineer, technical product manager, professional services engineer and entrepreneur.
- Technical expertise in object-oriented analysis & design, distributed computing, web services, enterprise integration, UI/UX design and cloud services & deployments.
- Exceptional CS education and expert knowlege of CS fundamentals, algorithms, data structures, asymptotic complexity, discrete mathematics and probability.
- Business acumen and the ability to understand the business impact of technology. I've held client-facing roles and have experience using concrete skills like active listening and relationship building to create harmony & rapport between product, sales, and engineering stakeholders.
- Experience assembling and managing cross-functional teams, elucidating timelines and deliverables, and making educated trade-offs on what can be delivered on-time and in-budget.

experience

Captricity, Inc., Oakland, California
Professional Services Engineer, 11/2014–10/2015

Captricity is a machine-learning startup that uses crowd-sourced deep learning to digitize paper documents—including those with human handwriting—at greater than 99% accuracy. Customers are large enterprises like life & health insurers and government agencies with massive amounts of data locked in paper.

- Designed and built secure, high-capacity integrations that connected the EDM systems of enterprise clients to Captricity's digitization pipeline. Leveraged cloud-based services for messaging and data transformation, including Mulesoft and AWS S3, EC2, RDS, EBS and Lambda.
- Engineered integrations that included complex, compute-intensive steps, like running computer vision and statistical learning algorithms to recognize one or two pages of a given kind—such as a birth certificate or prescription record—from among several hundred unwanted pages.
- Designed and managed scalable integrations that increased performance dynamically as client data increased. Built a parallel, distributed-memory compute cluster in the cloud that scaled from 1 to 16 GPU worker nodes as needed. Cluster workloads were computer vision heuristics for document recognition.
- Built integrations subject to stringent security and reporting requirements like HIPAA and California Civil Code §1798.82. This required careful consideration of issues like multitenancy and data isolation.
- Held a client-facing role. Met directly with customers to perform business process analysis, capture requirements, formalize artifacts and deliverables, and design and author solution proposals.
- Designed and built an object framework that enabled engineers throughout Captricity to easily assemble ACORD-standard messages in-memory, validate them, and serialize them to XML.

Unnamed Stealth Mode Startup, Fremont, California
Founder & Principal, 8/2013–9/2014

Worked to launch a startup offering a new approach to online dating. Novel features included harvesting users' location and calendar information and combining it with restaurant reviews and hours, allowing my site to suggest dates to users—complete with day, time, partner and venue. Even if you're a little shy, it's hard to say no to a date when a cloud server makes all the arrangements for you. This lowered the barrier to getting users on dates, providing an advantage over existing sites where it's all too common for people to be matched but never meet in person.

- Employed an API-first design in which a JSON REST API was conceived and specified before any consideration was given to client UI/UX.

- Prototype clients were developed for desktop and mobile platforms. The desktop client was an in-browser single-page application; the mobile client was an installable app for iOS devices.
- The in-browser client used AngularJS for data binding and made heavy use of HTML5 semantic elements, controls, and the canvas element along with CSS3 animation to provide rich transitions, overlays, and interactions.
- The backend was implemented as POJO Java Servlets that connected to any JPA data source. I made the choice to use Servlets over alternatives like Python+Django or Ruby+Rails because my backend dealt only in RESTful CRUD transactions and served no presentation information. This made the templating facilities of Rails and Django unnecessary. With Java, I enjoyed use of the excellent Google Gson library for marshalling and unmarshalling JSON and the unmatched scalability of the JVM and the JPA ORM framework.
- Explored options for deployment on and became deeply familiar with various cloud platforms, including Amazon AWS, Google App Engine, and IBM Bluemix.
- Developed a lightweight REST client framework that I used to harvest calendar information from the Google Calendar, Outlook.com, and Facebook Events APIs as well as restaurant and bar reviews from the Yelp and TripAdvisor APIs. I made the decision to develop my own framework because I found existing Java REST frameworks (e.g. Apache Jersey) too heavyweight for my needs.

Yorba Foundation, San Francisco, California
Senior Software Engineer & Product Manager, 4/2011–5/2013
Software Engineer, 6/2009–4/2011

Yorba was an open-source nonprofit whose mission was to bring tools for digital media to the Linux desktop. Photos, music, and videos are the stuff of peoples' lives—but tools for working with them on the open desktop lagged behind Windows and OS X. Yorba sought to change this, and up to the point the foundation was defunded in 2013, enjoyed many successes. Shotwell—the product that I managed—lives on as a project of the GNOME Foundation. My personal contribution to the project is evident by browsing the commit logs on gnome.org.

- Served as tech lead for Shotwell, the most widely used photo manager on the Linux desktop. Shotwell ships out-of-the box as the default photo application on both Ubuntu and Fedora and is used by over half-a-million people.
- Provided product vision and leadership. Made feature decisions. Designed user interfaces and storyboarded user interaction flows.
- Wrote some 25,000 of Shotwell's ≈115,000 lines of C and Vala code.
- Made key architectural decisions and reviewed every changeset.
- Managed and mentored junior engineers inside Yorba as well as outside contributors through Google's Summer of Code program.
- Designed and implemented a RESTful web service client framework with support for the full spectrum of HTTP transactions, XML processing, and OAuth authentication. My framework enabled Shotwell users to publish photos to social web services like Facebook, Flickr, and Picasa Web Albums and later evolved into an API that allowed outside contributors to write plugins for less-common services.
- Designed and implemented other major Shotwell features, including color, exposure, and tone adjustment, one-click auto-enhance, and photo printing.
- Wrote code for other projects as needed, including the for-fee consulting projects that paid Yorba's bills. This work allowed me to renew my C++ expertise, learn the Qt user-interface framework, and write richly animated user-interface code for multitouch devices.
- Represented Yorba at international conferences in Berlin, Montréal, and The Hague.

academic projects & research

- Developed a novel interactive digital painting system that uses concepts from texture synthesis and image processing to achieve style-adaptation, or the ability to simulate a wide variety of artistic styles and media via a single stroke rendering algorithm. The system is implemented in C, C++, and GLSL and makes significant use of modern, programmable graphics hardware. A preliminary write-up and source code are available at www.lucasbeeler.net/software/studio580/.
- Designed and implemented a parallel, distributed memory, Monte Carlo ray tracing system. When run on a parallel supercluster at the University of Michigan's CAC supercomputing center, linear or near-linear speedups were achieved as the application scaled from 2 up to 32 compute nodes. The system is implemented in C and uses the HPC standard MPI library for inter-node communication. See www.lucasbeeler.net/software/mpirt/1/ for a write-up and source code.

education

University of Michigan, Ann Arbor
B.S. & B.S.E. in Computer Science, 2008
Cumulative GPA **3.3 / 4.0**

technical skills

- **Languages:**
  Java, C, C++, Vala, Python, JavaScript, Objective-C, C#, XML, SQL
- **Operating Systems:**
  Extensive knowledge of the UNIX/Linux command line, including shell scripting, piping, regular expressions, and lightweight virtualization (e.g. `chroot`'ed environments and zones)
- **Cloud Platforms:**
  Amazon AWS—EC2, S3, S3 events, EBS, RDS, MTurk, Lambda. **Google App Engine**—App Engine Platform, Datastore, Objectify, App Engine JPA interface. **IBM Bluemix**—Liberty for Java, DB2 on Cloud
- **Server-Side, Web Services, and SOA Technologies:**
  RESTful web services, Java Servlets, Apache Tomcat, JDBC, JDK 1.5+ concurrency primitives, HTTP internals, XML DOM object-binding APIs (e.g. JAXB, PyXB)
- **Distributed Computing:**
  MPI, Redis, JMS (implemented as Apache ActiveMQ)
- **Unit, Functional, and Performance Testing:**
  *x*Unit (JUnit, Python unittest, CppUnit), Selenium, JMeter, Oracle VisualVM
- **Client-Side, Technologies:**
  AngularJS, jQuery, Bootstrap, AJAX, standards-compliant HTML5 & CSS3, HTML & XML DOM processing
- **Databases:**
  MySQL / MariaDB, PostgreSQL, IBM DB2
- **Desktop Technologies & APIs:**
  Gtk+/GNOME, OpenGL, WebKit, Win32 SDK, Qt/QML, Java AWT & Swing

professional organizations

- Vice President, University of Michigan ACM SIGGRAPH student chapter, 2006-2007
- Member ACM 2006-present
- Voting Member, GNOME Foundation, 2010-2013