

# Shane Holloway

url: [shaneholloway.com/resume](http://shaneholloway.com/resume)  
updated: Jan 2018 — PDF format  
email: [shane.holloway@ieee.org](mailto:shane.holloway@ieee.org)  
tel: +1(719) 237-0682

[LinkedIn](#) [StackOverflow](#) [GitHub](#) [BitBucket](#) [SpeakerDeck](#) [Twitter](#)

**Objective:** to deliver world-class products that astound customers in function and form, as leader of a dynamic, sharp, engaged, and focused team.

Entrepreneurially minded, diversely talented, and driven software engineering professional. Extensive experience in software product engineering, leadership, management, development, deployment, and operations. Specializing in isolating development and change risks through a hybrid of component, service, and object-oriented architecture, backed by 21 years of professional experience. Expert know-how and continuing interest in asynchronous and distributed computing, network communications, library factoring and abstraction for component reuse. Adept at leading and managing development teams, as well as contributing at a principal engineer level. The ideal position would start with a challenging problem that will improve the world, combined with a smart, energetic team that has the freedom to make it happen.

## Areas of Interest

- Architecture and implementation of software systems
- Distributed systems, actors, concurrency, and message passing
- Object, functional, and component oriented software
- Dynamic languages and meta-programming
- Visualization and graphics
- Continuous Integration, Testing, and Deployment
- Entrepreneurship, measurement, and marketing
- Technical management, coaching, and mentorship
- Business roadmap, vision, and planning

## 2016 - 2018 Chief Development Officer, Principal Engineer at ScribeRule, Inc.

In the Chief Development Officer role, I worked with the CTO and CEO to establish our product vision, direction, platform, and high-level requirements. Using a kanban-based agile development process, our 7-person technical startup team defined and built:

- a microservice-based distributed backend on Docker and Swarm
- a Microsoft Word add-in using .NET
- a collaboration app for use on mobile, desktop, and web-based use in ReactJS
- a CI/CD workflow using Gitlab-CI, AWS S3, and Docker Swarm
- a scalable UI-first testing platform for automated end-to-end validation

In the Principal Engineer role, I developed ~90% of the company's core intellectual property, defined the cryptographic data security architecture, and the backend microservice architecture. I built the combined continuous build and interactive development tooling for our Docker-based microservice architecture, improving developer productivity, lowering difficulty of developing docker services, and simplifying debugging.

## 2013 - 2016 Software Development Manager (SDM) at Amazon Web Services (AWS) / Amazon.com

As SDM, I lead two teams of software engineers (SDEs), owning 8 internal developer productivity tools, including multiple code browser, code review, code search, and code sharing products. All products operated at 99.9+% availability, with deep operational scrutiny for each outage. Targeting enterprise customers, we prioritized security, operational monitoring, workflow integrity, integration with other Builder Tools and company systems, and customer experience. Products include website and command line user interfaces, multiple supporting services, and AWS databases.

As SDE for the initial 5 months, my team delivered a popular company-wide project bootstrapping tool, helping customers focus on building their ideas by setting up all the infrastructure systems on their behalf. Over 20K projects and 100K shared libraries have been created with the tool.

TALKS: [HTML5 Platform — a 10K Foot View](#) on July, 2013 at [CS OSS Meetup Group](#)  
[Meta: Programming, Objects, and Classes](#) on April, 2012 at [CS OSS Meetup Group](#)

OPEN SOURCE: [py TG.kvObserving](#)  
[py TG.metaObserving](#)  
[py TG.quickSilver](#)  
[py TG.geomath](#)  
[py TG.helix](#)  
[node late](#)

STACK: **JavaScript** — over 10 years  
**Python 2** — over 15 years  
**C/C++** — over 15 years  
**Platforms** web, desktop, server, docker, embedded  
**HTTP/S/2** CORS, CSP, STS, Caching  
**HTML5** DOM, WebSockets  
**CSS3** CSS-in-JS, LESS, SASS, Stylus  
**APIs** OpenGL, libuv, NumPy, SciPy  
**AWS** Lambda, SNS, SQS, SWF, DynamoDB, S3, RDS, IAM, CloudWatch  
**Database** SQLite3, PostgreSQL, MySQL, DynamoDB, LevelDB, Redis, CouchDB, RethinkDB, ArangoDB

KIT: **SCM** Mercurial, Git, SVN  
**Editors** VIM, Atom, MS Visual Studio  
**Server** Docker, Nginx  
**Server-JS** Node.js, Express, Connect  
**Server-Py** Flask, uwsgi,  
**OS** Mac, Linux, Windows, iOS

INFO: **Microservice backend;** on-site or centralized installation.  
**4 customer-facing products:** desktop, web, mobile

STACK: **Lang 1st** JavaScript, Babel, ES6+  
**Lang 2nd** Python 2.7, 3.5, Bash  
**Web** ReactJS, Semantic-UI, Express  
**App** .NET, Electron, React-Native  
**DBs 1st** ArangoDB, CouchDB  
**DBs 2nd** LevelDB, PostgreSQL  
**DBs 3rd** RethinkDB, Redis

KIT: **AWS** S3, DynamoDB, Route53  
**Docker** 1.12, 18.01, Swarm  
**CI/CD** rollup, webpack, Gitlab-CI, multi-stage dockerfile  
**Hosting** Digital Ocean, Vultr  
**Linux** Debian, Ubuntu, Alpine  
**Windows** 7, 8, 10  
**Services** Prometheus, Grafana  
**Editors** VIM, Atom  
**SCM** Mercurial, Git

INFO: 8 internal products  
CodeCommit Console features  
7K-25K internal customers  
\$12K/month server spend  
99.9+% availability  
36 business goals

TEAM: 2 SDE teams  
2 SDE promotions, 3 hires, 2 exits  
4 SDE interns hosted

KIT: **Web** HTML5, ES6, CSS3, LESS, AngularJS, jQuery, Ruby, Rails, Node.js  
**AWS DB** DynamoDB, S3, RDS MySQL, RDS PostgreSQL  
**AWS** Lambda, SNS, SQS, SWF, IAM, CloudWatch, CloudTrail  
**Linux Server** Apache, RHEL5, ALLinux  
**Tools** Git, VIM, tmux

## 2009 - 2013 **CEO, Entrepreneur, Software Engineering Consultant, Co-owner** at *TechGame Networks*

### ENTREPRENEURIAL

In 2009, I returned to consulting to support a dream of creating and selling niche software products and services. Time was split between the needs of consulting clients and the multifaceted roles of entrepreneurship. Several product ideas faltered; some for lack of understanding, others due to partnership issues.

In 2012, I created [Bellite.io](#) as a software toolkit to help developers and web designers deliver hybrid desktop applications on both Windows and Mac OS X using tools they are already familiar with. (Modern examples include [Electron](#) and [NW.js](#).) After soft-launch in August 2012, I spent significant time adapting the product to customer needs and intensive learning, planning, and execution on marketing. New skills include applied analytics, metrics, A/B testing, email marketing, retargeting, advertising, IT operations, payment processing, email notification, development and administrative outsourcing. Applying these newly learned skills, I relaunched Bellite.io on the Node.js mailing list in June 2013. Later in June, to my misfortune, Google unexpectedly announced the end-of-life of Chrome Frame, which is a core component of the architecture.

**STACK**  
**Lang 1st** Python 2.7, JavaScript, CoffeeScript  
**Lang 2nd** C, C++, Objective C++  
**Web** HTML5, CSS3, Stylus  
**GUIs** Cocoa, Win32, WTL, ActiveX  
**Misc** JSON-RPC2, libuv

**KIT**  
**Server** Nginx, Node.js  
**Linux** Ubuntu, Arch  
**Windows** XP, 7, 8  
**Mac OS X** 10.6, 10.7, 10.8, 10.9  
**Editors** VIM, Xcode, MS Visual Studio 2010  
**SCM** Mercurial, Git  
**Services** Stripe, Mandrill  
**Analytics** KISSMetrics, MixPanel  
**Pixels** Pixelmator, ScreenFlow, Motion

## 2009 **Systems Engineer** at *RT Logic*

As Systems Engineer at RT Logic, I contributed stability and performance improvements on a hardware product performing real-time simulation of RF signal delay and distortion for testing signal conditions encountered during aircraft, satellite and missile flights. In my second project, I assisted and supported the deployment and testing of a satellite interference geolocation system that allows operators to find the source of accidental or purposeful signal and bandwidth interference. I contributed to project design, team tool support, and adapting the software to work with RT Logic hardware from other departments. My final project involved creating a network status and control protocol in embedded vxWorks using C/C++ to interface with a new FPGA-based PCI card for high-speed signal digitization.

**STACK**  
**Lang** C++, C#, Python 2.7  
**GUIs** .NET  
**Formats** JSON, XML, proprietary  
**Protocols** RS-485, RS-232, TCP & UDP  
**Libs** Poco, NumPy, SciPy  
**DBs** MySQL

**KIT**  
**SCM** SVN, Git, Mercurial  
**Editors** MS Visual Studio 2008, VIM  
**Windows** XP, Vista, 7  
**Windows Server** 2003, 2008  
**vxWorks**— embedded PCI to FPGA

## 2005 - 2009 **Lead Software Engineer** at *AbleLink Technologies*

Lead developer of more than 10 Phase Is, 5 Phase IIs, and 2 FastTrack research grants for the Department of Education and the Department of Health and Human Services focusing around the topic of technology accessibility for the cognitively disabled. SBIR grants help fund high-risk project research and development from small businesses, and are generally very fast-paced software engineering projects with research-vetted results. During the second phase of these research grants, we transformed preliminary studies into feature rich commercially-available products tailored for the cognitive disability market. Combining the gamut of delivery platforms with the ease of use required for our primary cognitive disability audience proved very challenging and rewarding, fostering expertise in a wide variety of languages and technologies from cloud to embedded platforms.

**STACK**  
**Lang 1st** Python 2.4, 2.5, 2.6, 2.7  
**Lang 2nd** C, C++, JavaScript, Smalltalk, ActionScript  
**GUIs** wxWidgets, QT4, QT5, OpenGL  
**Web** HTML5, CSS3, Jinja2  
**DBs** SQLite3, QuicksilverDB, ZODB  
**Services** AWS S3  
**Libs** NumPy, JQuery, OpenAL  
**Formats** Pickle, JSON, XML

**KIT**  
**Windows** XP, Vista, 7, 8  
**Windows Mobile** 2003, 5, 6  
**Mac OS X** 10.6, 10.7, 10.8  
**iOS** 3, 4, 5  
**Tools** Parallels  
**Editors** VIM, MS Visual Studio 2003  
**SCM** Mercurial, SVN

## 2003 - 2005 **Software Engineering Consultant, Co-owner** at *TechGame Networks, LLC*

Launched the company with two partners with the intention of developing software and systems products. During this time frame, my primary role was executing on consulting services for clients. I continued working with Redstone to perform custom engineering and integration services for the City of Los Angeles security center and the Ontario International Airport in California. Consulted for Agilent Technologies to assist with C++ threading integration problems, as well as a new OpenGL rendering components. Began consulting for AbleLink Technologies, assisting their SBIR research grants for helping people with intellectual disabilities through software.

**STACK**  
**Lang 1st** Python 2.3, 2.4  
**Lang 2nd** C, C++, Smalltalk  
**GUIs** wxWidgets, OpenGL  
**DBs** SQLite3  
**Formats** XML, JSON

**KIT**  
**Windows** 2000, XP, Vista  
**FreeBSD** 5, 6, 7  
**Editors** VIM  
**SCM** SVN

## 2002 - 2003 **Principal Systems Engineer** at *Redstone Integrated Solutions*

Led the team that architected, designed, implemented, and delivered a flexible and customizable security monitoring and control application. This system was then customized and integrated into the police operations center of the Ontario International Airport in California. As configured, the system monitors and controls 30 cameras simultaneously, allowing switching among 200 cameras, manages a sensor network of over 3,000 alarm points, and enables centralized point-and-click control of field devices such as door locks.

**STACK**  
**Lang** Python 2.2, C, C++  
**GUIs** wxWidgets, ActiveX, OpenGL  
**DBs** SQLite3, OODB  
**Protocols** RS-232, TCP & UDP  
**Formats** XML, SVG

**KIT**  
**Windows** 2000, XP  
**Editors** VIM  
**SCM** SVN

2001 - 2002 **Software Engineer** at *ICCE Technologies*

Co-architected a distributed software system to provide access, control, and event propagation for geographically distributed hardware. My primary role was framework development, including application skinning, Jabber communications, services, and automation toolkits. The flagship ICCE Vision Application was built upon the application toolkit framework, enabling a fully-customized, cross-platform skin user interface, communicating to distributed hardware over Jabber.

**STACK**  
**Lang** Python 2.1, 2.2, C, C++  
**GUIs** wxWidgets  
**Protocols** XMPP, RS-232, TCP & UDP  
**Formats** XML, proprietary  
**KIT**  
**Windows** 98, NT 4, 2000, XP  
**Server** Jabber, Linux  
**Editors** VIM  
**SCM** SVN, CVS

1996 - 2001 **Software Engineer** at *Decision Science Applications (SM&A, Emergent)*

Contributing developer on more than 12 successful HP/Agilent Technologies products, including design and implementation of two interactive scientific graphing components, interaction-based calculation algorithms, and infrastructure design and construction. During my capstone project at Emergent, I served as both a software architect and a key developer for Agilent MUI, including system design, framework and infrastructure, data management, algorithmic development, and graphical interfaces.

**STACK**  
**Lang** C, C++  
**GUIs** MFC, WTL, ATL, ActiveX, OpenGL  
**Protocols** HPIB, GPIB, RS-232  
**DBs** MS SQL Server, MFC/OLE  
**KIT**  
**Windows** 3.1, 95, 98, NT 4, 2000  
**Editors** MS Visual Studio 6  
**SCM** MS Visual Source Safe

1998 - 2000 **Bachelors of Science in Computer Science** at *University of Colorado*

Minor in Mathematics. Minor in Economics.  
First in Class. (GPA: 3.982; GRE: 790/800/640)  
Outstanding Undergraduate Student from the College of Engineering and Applied Science.

**STACK**  
**Lang** LISP, Scheme, Pascal, C, C++  
**Network** BSD Sockets, TCP  
**Graphics** OpenGL, DirectX 1