|  |  |
| --- | --- |
| Brian LauberSoftware Shepherd | * 1820 Swann St. NW, Apt. 404 Washington, DC 20009 * 216-394-2081 * [blauber@arclia.com](mailto:blauber@arclia.com) |

### Summary

I am an experienced Software Engineer with a passion for guiding organizations toward success. I specialize in dispelling uncertainty and cultivating a shared understanding across teams.

|  |  |
| --- | --- |
| Personal Site | <https://brianlauber.com/> |
| GitHub | <https://github.com/briandamaged/> |
| LinkedIn | <https://www.linkedin.com/in/brian-lauber-26197919/> |

### Professional Skills

|  |  |
| --- | --- |
| Leadership | Project Management, Requirements Gathering, Road-mapping, Documentation, Code Reviews, Process Improvement, Software Architecture |
| Code Paradigms | Object-Oriented Programming, Functional Programming, Metaprogramming, Domain-Specific Language Design |
| Languages | Python, Javascript, Typescript, Ruby, Java, C#, C++, C, Bash |
| Frontend Tooling | React, Next.js, Gatsby, WebPack, Babel, MDX, Tailwind CSS, PostCSS |
| Web Service Tooling | Koa.js, Express.js, Flask, Django, Rails, Sinatra, REST, SOAP, Celery, Resque, OpenAPI |
| HTTP Servers | Nginx, Apache httpd, Apache Tomcat |
| Databases | MySQL, PostgreSQL, MSSQL, MongoDB, Redis, Hive, RabbitMQ |
| DB Query Builders | SqlAlchemy, Knex.js, LINQ, Hibernate |
| ORMs | SqlAlchemy, ActiveRecord, Mongoid, MongoMapper, Mongoose, Bookshelf.js, Sequelize, Django (models), LINQ, Hibernate |
| Data Science | NumPy, Pandas, scikit-learn, ray, Jupyter, MLFlow |
| Automated Testing | Pytest, unittest (Python), RSpec, Junit, MochaJS, Chai, Selenium, Capybara, QTP, LoadRunner |
| Static Analysis | SonarQube, Fortify SCA |
| Security | Hardware Security Modules, PGP, Cryptography, HashiCorp Vault |
| Version Control | Git, Subversion, CVS, StarTeam |
| Dependency Mgmt. | Pip, Poetry, VirtualEnv, Gem, Bundler, rvm, npm, yarn, nvm, Maven |
| Build Tools | Babel, WebPack, Rake, Ant, Maven, Make, Paver, setup.py, Poetry, Packer |
| Containerization | Docker, Podman |
| Virtualization | Parallels, VirtualBox, VMWare Player, Packer |
| Deployment | Terraform, Kubernetes, Capistrano, Fabric, Heroku |
| CI/CD | GitLab-CI, Jenkins, Hudson, Cruise Control |
| Cloud Platforms | AWS, Azure |
| Issue Tracking | JIRA, Redmine, Trello, Asana |
| Collaboration | Slack, Discord, Teams, Google Suite, Microsoft Suite, Markdown, draw.io, Gliffy, Whimsical, Mermaid, QuickTime, Confluence |

### Professional Experience

#### Plutometry Corporation (October 2019 – Present)

Plutometry utilizes machine learning to make financial predictions for their customers within a highly-secured environment. I served as the Lead Developer for most of their applications, working primarily in Python, Java, and C.

##### Principal Software Engineer *(October 2019 – January 2024)*

I was initially recruited to implement a proprietary Blinding Token technology on a Hardware Security Module. Afterwards, I served as the company’s Lead Developer.

* Delivered Plutometry’s first revenue-generating platform. (Defined the scope, gathered requirements, established roadmaps, collaborated across teams, coached engineers, implemented key components, provided operational support, etc.)
* Reduced the end-to-end execution time of Wealthpoint by a factor of 200, thus enabling Plutometry to conclude its pilot program and graduate to a national rollout.
* Established Project Management practices. (Setup JIRA, trained staff on agile methodologies, served as the scrum master)
* Promoted SDLC Best Practices. (Test coverage, code reviews, continuous integration, release management, internal libraries, private package registries, documentation)
* Defined the Mathematical Underpinnings for Plutometry’s Machine Learning Models. (Standardized terminology, untangled key concepts, defined mathematical operations)
* Provided key insights that often reshaped the way the research team conceptualized things. (Ex: demonstrated ways to quantify the topological characteristics of the models, which ultimately helped the team understand some counterintuitive observations.)
* Delivered a new “Model Laboratory” platform that empowered the Research Team to train and test models on their own without requiring assistance from the Engineering Team.

##### Principal Software Engineer *(June 2024 – Present)*

I returned to Plutometry on a temporary basis to help them navigate through a critical period and train their new engineers. As such, my work during this brief period was fairly limited in scope.

* Repaired the infrastructure that was critical to product delivery. (AWS, Azure, Vault, Kubernetes Clusters, GitLab, Nexus, SonarQube, and more)
* Guided Customers through operational workflows and ensured that their financial predictions were delivered on schedule.
* Authored onboarding procedures for new team members.
* Trained new team members.

#### 33Across (May 2016 – October 2019)

33Across maintains infrastructure that delivers online advertisements to web sites. During my tenure, the business sought to create a new platform that delivered advertisements using a modern technique known as “Server-Side Header Bidding”. I was the technical lead for this initiative and many others, which demanded expertise in both JavaScript (Node.js) and Ruby.

##### Principal Software Engineer *(May 2016 – October 2019)*

I was initially recruited to help 33Across modernize the UI for their Unified Dashboard. Soon afterwards, I was promoted to Principal Software Engineer and served as an advisor across numerous projects.

* Architected a globally-available advertisement delivery platform that served millions of requests per minute with a target latency of 5ms per request.
* Promoted SDLC Best Practices. (Test coverage, code reviews, continuous integration, release management, internal libraries, private package registries, documentation)
* Coordinated the delivery of numerous software projects. (Gathered requirements, defined scope, established roadmaps, collaborated across teams, coached engineers, implemented key components, etc.)
* Designed and implemented several experimental technologies. (Tooling for syndicating the rollout of advertisement preferences across platforms, specialized pattern matchers, declarative languages for policy and enforcement)

#### Arclia LLC (July 2014 – Present)

I launched Arclia LLC so that I could take on contract work from local startup companies. My contracts with 33Across and Plutometry quickly converted into full-time employment.

##### Owner + Principal Software Engineer *(July 2014 – Present)*

* Guided Startup Founders toward clear Product Visions.
* Developed Prototypes and Minimum-Viable Products that could be shopped around to potential Customers.
* Adopted codebases and improved their code quality. (Tackled performance bottlenecks, refactored code for maintainability, mitigated security issues, automated deployments, renovated user interfaces)
* Established Project Management practices. (Introduced companies to JIRA and other solutions, trained staff members on agile methodologies, served as the scrum master)
* Introduced SDLC Best Practices. (Test coverage, code reviews, continuous integration, release management, internal libraries, documentation)

#### JIBE (June 2013 – June 2015)

JIBE significantly improved the user experience for job applicants without forcing companies to switch to a new Applicant Tracking System. Behind the scenes, this meant that the integration engineers had to invent clever ways to automate legacy software that was never designed to be automated.

##### Senior Automation Engineer (June 2013 – June 2015)

I primarily focused on re-architecting the JIBE Apply backend for long-term maintainability, reliability, and scalability.

* Instilled a culture of code reuse. (Encouraged integration engineers to create small libraries that solved specific problems. Worked with IT to provision a private package registry.)
* Standardized core infrastructure across customer integrations.
* Provided architectural oversight.
* Improved the efficiency of the integration engineering team by over 1000%.
  + Authored numerous libraries that encapsulated techniques for reliable UI automation.
  + Wrote a compiler that emitted all valid workflows for each client. This enabled the engineers to write a single high-level specification for each client rather than maintaining hundreds of individual workflows by hand.
* Coached junior team members.
* Established code review practices.

#### Cigital Inc. (June 2007 – June 2013)

I performed numerous different roles at Cigital, most of which were either directly or indirectly related to the security industry.

##### Senior Software Engineer (January 2012 – June 2013)

* Worked with customers to define product roadmaps.
* Served as the technical project manager for offshore teams.
* Revitalized a faltering product through rapid iteration and feature innovation. (This also repaired a customer relationship in the process!)
* Designed components that integrated static analysis into CI/CD pipelines. (Note: our team was pioneering these integrations before they became commonplace)

##### Quality Assurance Lead (January 2011 – December 2011)

* Established infrastructure for managing and tracking manual test cases.
* Provided technical support for customers.
* Coordinated onshore and offshore teams.

##### Security Consultant (January 2010 – December 2010)

* Conducted security reviews of customer codebases.
* Delivered security training to customers.
* Defined best-practices for security-oriented QA.
* Trained consultants on security-awareness in Rails.

##### Quality Assurance Automation Engineer (June 2008 – December 2009)

* Designed mutation-based testing tool for simulating user behaviors. (This uncovered numerous bugs that we had never thought to test for explicitly)
* Designed meta-programming language for QTP. (This enabled the QA team to write resilient automated tests that could accommodate changes in the UI)
* Trained QA teams on QTP / Quality Center / LoadRunner.
* Established Manual → Automated Test Case promotion workflow.
* Constructed distributed cluster of QTP machines.
* Deployed and managed QA infrastructure.

##### Software Intern (June 2007 – August 2007)

* Served as the lead developer for a reverse-engineering tool.

#### Air Force Institute of Technology (AFIT) (June 2005 – August 2006)

##### Software Intern (June 2006 – August 2006)

* Implemented a stochastic analysis program that computed probabilities for various types of failures within complex networks.
* Discovered a flaw in one of AFIT's algorithms. This flaw could cause the algorithm to produce probabilities that were outside of the range [0, 1].

##### Software Intern (June 2005 – August 2005)

* Translated a stochastic analysis program from Fortran to C++.
* Extended the program to allow for arbitrary probability curves and precision.

### Education

* Bachelors of Science in Mathematics (Case Western Reserve University, 2008)

### Side Projects

Most of the libraries that I’ve written are proprietary, but I managed to open-source some of the smaller libraries over the years.

|  |  |  |
| --- | --- | --- |
| **Title** | **Language** | **Purpose** |
| arclia-stats | Python | Implements weighted quantiles. (Functionality that is surprisingly absent from numpy!)  <https://github.com/Arclia/python-arclia-stats> |
| zxc | Javascript | Provides easy command-line access to project-specific executables within Node.js projects. This was a lot more useful before npx was invented!  <https://github.com/briandamaged/zxc> |
| Logbert | Ruby | A categorical logger for Ruby. This became the de-facto logger at JIBE because none of the other loggers fit their needs at the time.  <https://github.com/briandamaged/logbert> |
| Tolerable | VBScript | Leverages runtime code generation to add anonymous functions, inheritance, and meta-programming to the VBScript language.  <https://github.com/briandamaged/tolerable> |
| Unobservable | Ruby | Extends Ruby’s object model to support .NET-style events.  <https://github.com/briandamaged/unobservable> |
| B-Lazy | Ruby | Extends Ruby’s Enumerable mixin to support lazy evaluation. Ruby 2.0 incorporated something similar into its core API, thus eliminating the need for this library.  <https://github.com/briandamaged/b-lazy> |

### Hobbies

|  |  |  |  |
| --- | --- | --- | --- |
| * Bicycling * Hiking * Kayaking | * Cooking * Baking * Beer Brewing | * Karaoke * Piano * Songwriting | * Board Gaming * Video Gaming * Programming |