

Gareth Ari Aye <aria@caa.columbia.edu>
Website: *aye.science*
Phone: (206) 330 - 9782

Education

MS Computer Science - Machine Learning, Columbia University (2017 - 2019)
Python, TensorFlow, Keras, Theano, NLTK, sklearn

- GPA: 3.6
- Relevant coursework: Deep Learning, Machine Learning, Natural Language Processing, Computer Vision, Artificial Intelligence, Probability & Statistics, Cloud Analytics

BA Computer Science, Middlebury College (2008 - 2012)
C, Java, Assembly, Python

- Honors: Cum Laude, High Honors in Computer Science
- GPA: 3.52, Major GPA: 3.82
- Worked regularly as a teaching assistant and systems administrator in the cs dept

Work Experience

Senior Software Engineer, Google (2017 - 2019)
Python, TensorFlow, Keras, Java, IntelliJ Plugins, Dart

- Founded, led, and launched an effort to improve ranking in Dart code completion through language modeling (*wiki page*). This was the top announcement in Dart's major version release 2.5 (*VentureBeat*).
- Submitted *Sequence Model Design for Code Completion in the Modern IDE* to ICSE 2020 (*preprint*).
- Made a number of enhancements to Dart code completion across editors including IntelliJ such as suggesting symbols from not yet imported libraries and including package info in code completion suggestions.
- Built internal Dart language usage metrics pipeline and dashboard to help quantify impact of improvements to developer tools
- Implemented project-scoped cost management in GCP billing reports giving billing account administrators the granular ability to share costs associated with single GCP projects within their organizations. My work spanned GCP's IAM service, through cloud billing's web services and frontend permissions model, all the way to creating a new billing overview in GCP's project dashboard.

- Created time granularity controls for GCP's billing reports UI and API allowing cost viewers to aggregate and visualize their usage at different levels of specificity.
- Wrote a client-side cache for billing data to make GCP's billing reports UI faster for our biggest users by preventing the same usage data from being fetched multiple times while customers explore their cost data.
- Contributed to overall product and code quality on an ongoing basis by using the product, filing bugs, and fixing lots of papercuts outside of my direct project focus when time allowed.

Managing Partner, SciDock LLC (2016 -)

Emscripten, Firebase, JavaScript, React, Redux, Flowtype, Node.js, Express, Jest

- Grew a web-based \LaTeX editor *LaTeXBase* from 0 to >10k monthly active users.
- Cross-compiled the `pdflatex` compiler from C to JavaScript with Emscripten to unlock offline capabilities via service workers.
- Implemented real-time, Google Docs style collaboration on top of `ot.js`.

Engineering Lead, Mozilla Firefox (2013 - 2016)

JavaScript, IndexedDB, iCal, CalDAV, Node.js, WebDriver

- Promoted twice to grow from an IC to leading a small engineering team.
- Represented Mozilla in the standards committee that develops iCal and CalDAV.
- Wrote a mobile calendar application distributed with Mozilla's mobile OS that synced with CalDAV servers like Google's and allowed users to perform CRUD actions on their events.
- Wrote a popular, open-source JavaScript library for syncing calendars and contacts used in Firefox OS.
- Automated running UI integration tests in Firefox OS.
- Led an effort to speed up Firefox OS builds by leveraging the concurrency and caching abilities of the Ninja build system created by Google's Chrome team.

Software Engineer, Airtime (2012 - 2012)

JavaScript, Flash, Python, MongoDB

- Built a frontend flash p2p video chat application that interfaced with Airtime's JavaScript and Python APIs.
- Implemented recording and sending video messages to Facebook friends.
- Embedded YouTube's video player within Airtime's flash stack to unlock watching YouTube videos with friends over video chat.

Research Assistant, Middlebury College Computer Vision Lab (2010)

C, Python, MATLAB

- Programmed a collection of high quality cameras and projectors to automate capturing high-quality stereo vision depth maps based on Scharstein & Szeliski's structured light paper.
- Developed a process for camera calibration using MATLAB.
- Wrote a collection of image postprocessing scripts using Python to help scale up our lab's processes for processing and packaging captured datasets.