

# Tor E Hagemann

[Google Search](#)[I'm Feeling Lucky](#)

Residence: SF Bay Area, CA USA || Contact: <https://linkedin.com/in/hagemt> or <tel:+1-262-370-9942> (Twilio to Cell)

**Career Purpose** Currently employed full-time (but open to new adventures) as an experienced *software engineer*. Most interested in positions offering opportunities for continued growth, where I can contribute value to customers directly, *exercise technical and interpersonal skills*, as well as expand knowledge through the union of computer science, mathematics, and humanity.

## Professional Experience

Job Title	Company: Location	Time Domain $\{f(t)\}$
<b>Software Engineer</b>	<i>Simpler Postage, Inc.</i>	$t \in [2021-11, \text{Current}]$
<i>dba. EasyPost</i> = The simple shipping API. Member of the USPS and core carrier teams, responsible for a majority of the company's revenue. We deliver rates, labels, tracking, and other services through sophisticated PC Postage integrations, the UPS (Digital Access Program) and FedEx/DHL/etc.		
<b>Principal Software Engineer</b>	<i>Socotra</i> : San Francisco, CA	$t \in [2020-05, 2021-11]$
<b>Staff, Senior Software Engineer</b>	(title structure changed in 2019)	$t \in [2018-11, 2020-05]$
<i>Socotra</i> is modern SaaS for insurance core systems. We enable fast go-to-market (weeks or months, not years) for new and existing lines of business. Our highly-available policy and claims management APIs are upgraded continuously, extremely configurable, and benefit from an engineering pedigree of building secure information-processing systems for complex endeavors. We support most forms of property and casualty (P&C) with highlighted competence in the area of vehicle/fleet management, as well as some forms of business-owner policy (BOP) and life insurance products. We scale to many thousands of API calls, relevant to hundreds of customer-defined and versioned products, in order to integrate our environments' tenants with external services. At Socotra, I have played a key role in dozens of features and brought many to production myself, where I am frequently referenced as a whole-system authority. I have honed many skill sets here, which are enumerated upon other pages.		
<b>Software Engineer II</b>	<i>Cisco Systems</i> : Boxborough, MA	$t \in [2015-01, 2015-08]$
<b>Software Engineer III</b>	(returned to SF Bay Area in Fall 2016)	$t \in [2015-09, 2017-08]$
<b>(on public developer APIs)</b>	(Cisco Spark, re-branded: Webex Teams)	$t \in [2017-07, 2018-11]$
Launched as strong individual contributor to Context Service and Care Assistant offerings, solutions that integrate natural language processing (NLP) into customer care and the Webex Teams platform via bots and subject matter experts. Later, I was hired onto the core team that designs and implements public-facing APIs, abstracting myriad internal microservices. Acting as a strong developer experience advocate, quickly became a respected authority on services that leverage various private APIs: exposing polished RESTful interfaces as well as the engine that processes events into webhook deliveries. Try our interactive documentation via our developer portal: <a href="https://developer.webex.com">https://developer.webex.com</a>		
<b>Server-Side Quality Engineer</b>	<i>Palantir Technologies</i> : Palo Alto, CA	$t \in [2013-07, 2014-10]$
Automated and performed hundreds of hours of manual testing, eliminating complicated and arduous $O(n)$ work-flows with $O(1)$ scripts; contributed targeted (commit level) inspection and verification of important bug fixes; vouched for functionality and usability of improvements to complex back-end features on mainline products, shipped weekly as patches to a large host of production systems.		
<b>Undergraduate Intern</b>	<i>Kitware Inc.</i> : Clifton Park, NY	$t \in [2011-06, 2011-12]$
Built maintainable scripts and pipelines, to provide early signals through CI/CD on two sophisticated scientific computing (Computer Vision) projects; leveraged <code>git</code> and <code>cmake</code> in a sensitive data context.		

## Technical Skills

These *gauge* my own ability to answer questions relating to the following areas:

- GoLang, gRPC+protobuf: simple CLI tools and/or easy-to-deploy, Dockerized executables
- Docker, incl. compose: I author and maintain local dev setups on teams of ~2-10+ engineers
- NextJS, paired with CDN and/or ngrok to build [multi “single page” applications](#) very quickly
- NodeJS: services in production + [several small packages](#) (likely: none will rise to Internet fame)
- Web “Frameworks” (React + friends, [X]HTML/CSS) *examples on [GitHub](#) and [my personal site](#)*
- Full-stack JavaScript (many NPM packages) *modern, event-based microservices* (Cloud/Hybrid)
- JVM Languages/Tools (Java, Kotlin, some others) *very strong, recent* (many build systems/IDEs)
- Scripting (Shell/Python, some Ruby/Perl) *advanced, regular practice* (prefer: EDITOR=vim {ba,z}sh)
- Version Control, advanced: we can git things merged; I’ve led study groups at multiple companies
- C/C++ (projects using Boost, Qt) *strong, historically well-practiced* (gcc toolchain, some C/F#)
- Distributed Systems (Cassandra, Redis, Kafka, ELK) *as DB, cache, message Q’s, search logs/etc.*
- Observable production stacks: APMs like New Relic AND my own Grafana, Prometheus, etc.
- OLAP like ClickHouse: extract, transform, load (ETL) tooling for analytical + batch processing
- NoSQL, incl. MongoDB: from inherited subsystems that I may still complain about on occasion
- Kubernetes, ★★★★★: I’ll tell you what CRDs are, but I don’t personally manage k8s clusters
- Research/Technical Writing (Markdown flavors, L<sup>A</sup>T<sub>E</sub>X) proficient, quick study, [fan of papers](#)
- Mobile Development (Android, some iOS via React Native, interaction design) familiar, entry-level
- Embedded Development (hardware hacking; e.g. Raspberry Pi, Arduino) amateur (hobbyist) level
- I like soldering irons + will troubleshoot networks/Wi-Fi myself; I don’t like waiting on IT tickets
- Dunning–Kruger is real; I know I’m not the best at everything, but I like to avoid being the worst

## Related Skills

- Team building: strong contributor to company culture (work = play; avoid+resolve conflicts)
- Tech leadership: mentoring interns/peers with high-growth mindsets; light project management
- Interviewing engineers: executive search panelist, also coordinated early FE/BE hiring pipelines
- Customer interactions: key participant on Zoom under certain business contexts, esp. re: security
- Project multi-tasking: able to time-share in “office hours” consulting alongside my planned work
- DevOps, front-end, SRE XO: not my forte, but I am proudly conversant in related best practices
- Documentation, communications: I write, but I’m prone to overuse both colons and parentheses
- Creative and artistic: I enjoy reading, using code to build *pretty-enough things that do the job*
- Cheerleader for agility, delivery, diversity, and I’ll begrudgingly admit I’m SCRUM certified

## Education Summary

*Rensselaer Polytechnic Institute* (RPI) 2009—2013 (left school to pursue better growth options)  
B.S. Math/Computer Science (dual-major program; completed overwhelming majority of courses)  
Dean’s List  $\wedge$  paid tutor for campus Advising and Learning Assistance Center: 4 semesters each  
Semester study-abroad: *Danmarks Tekniske Universitet* (DTU) in Lyngby, Denmark (Fall 2012)

1. Artificial Intelligence (taught by Sebastian Thrun and Peter Norvig, Dir. of Research at Google)
2. Machine Learning (taught by Prof. Andrew Ng; often explored further work, beyond assignments)
3. In addition to those *Stanford University* MOOCs, completed DataCamp AI, ML, and NLP courses
4. Coursework toward Udacity’s machine learning nano degree program (not looking to complete)