

Avi Flax

Software Developer

I design¹, build, document, and operate software systems.

▼ Objective

My ideal role would involve designing¹, building, documenting, and operating software systems in close collaboration as part of a cross-disciplinary team of peers, mentors, and mentees.

I can do either high-level or low-level work but I'm happiest when I can alternate between them.

In addition to working on software *systems*, I also find myself drawn to optimizing software *teams* — so it'd be great if that could be a small part of my role as well.

I'm open to a formal leadership role but ultimately I find bottom-up leadership more compelling.

▼ Current Position

2017–present

Principal Software Engineer at [Funding Circle](#)

- Driving an initiative to level up the documentation and knowledge sharing culture/practices of a ~250-person software product development group
- Taken point on designing and implementing a useful information architecture for the corporate wiki
- Conceived, developed, released, and maintain [the FC4 Framework](#), an open-source framework for authoring, publishing, and maintaining software architecture diagrams (includes a CLI tool written with Clojure)
- Working on improving organizational effectiveness; e.g. communication channels, practices, etc.
- Technologies, techniques, and tools employed:
 - [Clojure](#)
 - [Ruby](#)
 - [JavaScript](#) (with [Node.js](#))
 - [Docker](#)
 - [CircleCI](#)
 - [GitHub Actions](#)
 - [Docs as Code](#)
 - [Confluence](#)

▼ Prior Experience

2016–2017

Principal Software Architect at [Park Assist](#)

- Facilitating the company's transition from execution phase to growth phase
- Increasing collaboration and cohesion of the engineering team
- Establishing roadmaps for all systems
- Designing, building, and operating:
 - streaming data pipelines conveying data from sites around the world to a central data center
 - stream data topologies using [Kafka](#), [Kafka Streams](#), [Kafka Connect](#), [JRuby](#), and [Clojure](#) to effect high-throughput and low-latency data-centric applications
- Driving adoption of:
 - infrastructure-as-code, configuration management, and immutable infrastructure with [Terraform](#), [Ansible](#), [CircleCI](#), [AWS](#), and [Kubernetes](#)
 - consistent conventions around testing, documentation, code review, and continuous integration
- Operating services with [Terraform](#), [Ansible](#), [AWS](#), and [Kubernetes](#), [Datadog](#), [Slack](#)
- Defining and executing new approaches to:
 - engineering support
 - team structure
 - hiring
- Create a culture of caring about sustainability, documentation, and maintainability
- Introducing and championing new:
 - development tools: [GitHub](#), [Terraform](#), [CircleCI](#)
 - company-wide communication media: [Slack](#), [Discourse](#)
- Mentoring those new to software engineering

2015: Timehop

- Refactored a critical and complex system into a loosely-coupled stream-based system using Kinesis and Go ([slides](#))
- Designed and implemented sophisticated integration with Twilio for SMS-based signup
- Designed and implemented a large-scale data processing tool to process billions of records very quickly

2014: Thinkful

- Designed and implemented an event-driven system to automate Stripe subscription management

2013–2014: SFX Entertainment

- SFX acquired Arc90 in 2013 to be its internal app design and development group
- I initially adopted the title CTO, then recruited [someone amazing](#) to replace me and shifted to *Chief Software Architect*
- I helped design and build an ambitious new-from-the-ground-up streaming music platform and integrate it with existing products
- I helped shape the organization by helping craft its structure, practices, culture, goals, communication, and approach
- I recruited key personnel

2005–2013: Arc90

- Arc90 was a Web application design & development consultancy; it was acquired by SFX Entertainment in 2013
- Wore many hats: developer, architect, team leader, partner, director
- Worked with teams to design, build, and maintain sophisticated software platforms, including:
 - A Web/SOA insurance platform covering all key aspects of policy origination and management
 - A Web-based search engine and reference platform for engineers
 - An API at the heart of a new-from-the-ground-up news distribution platform for Reuters
- Conducted technical audits of clients' acquisition targets
- Helped manage the business as a partner
- Managed client relationships
- Invested much time and effort in recruiting and retaining exceptional team members
- Started practice of regular company-wide code review

2001–2004: ADP

- Refactored, enhanced, and maintained a sophisticated application for producing custom financial documents for on-demand printing

2001: Register.com

- Team lead position for large high-traffic auction site
- Responsibilities included designing, implementing, and maintaining features; and reengineering site technology and architecture
- Created new internal tools and development procedures

2000: RewardsPlus

- Maintained and enhanced a large-scale online employee benefits enrollment system for diverse clients with diverse needs

1998–2000: Words In Progress

- Developed requirements and specifications for high traffic websites directly with clients; crafted application architecture and database design

1998: Ideal Computer Strategies

- Worked with teams of designers, coders, and project managers to concurrently develop and deploy client websites with basic dynamic features

1997–1998: PCC Internet Design

- Founded and managed a small web design shop in Pikesville, MD providing full-service web design and development to small companies in the Baltimore area

▼ Values

I believe that:

...diversity and inclusion are crucial *and* create rich, healthy, sustainable, productive organizations.

...[DRY](#) and [the Rule of Three](#) apply to docs just as much as code.

...people can only do great work if they are healthy, growing, supported, and respected.

...all great processes have one thing in common: they are regularly reviewed and refined.

...the most important aspect of any technology is its potential to solve problems.

...technology is a phenomenally powerful tool that we humans have at our disposal to solve problems — if we so choose.

...we do some of our best thinking in collaboration with others.

...institutional/communal memory is critical to building sustainable organizations.

▼ Key Skills

- Problem analysis
- Software design, architecture, construction, deployment, and operation
- Deep expertise in the World Wide Web and HTTP
- Web API design & development
- Product design
- Data engineering
- Documentation, information architecture
- Team/organizational leadership
- Assessing and improving organizational effectiveness
- Organizational architecture

See also [Additional Skills & Technologies](#) below.

▼ Key Technologies

- | | |
|---|--------------------------------|
| • Amazon Web Services (AWS) | • Kubernetes |
| • Clojure | • Lambda (AWS) |
| • Docker | • Python |
| • Java | • Ruby |
| • Kafka | • Terraform |
| • Kinesis | |

See also [Additional Skills & Technologies](#) below.

▼ Additional Skills & Technologies

- | | |
|--------------------------------|----------------------------|
| • Behavior Driven Development | • Information architecture |
| • Browser APIs | • JavaScript |
| • Business analysis | • Markdown |
| • CoffeeScript | • MongoDB |
| • CouchDB | • Node |
| • Datomic | • OmniGraffle |
| • DB Design | • PostgreSQL |
| • Django | • Restlet |
| • DynamoDB | • SQL |
| • ElasticSearch | • Solr |
| • Flask | • Stripe |
| • Go (Golang) | • UX design |
| • Groovy | • XML/XSL/XSD/XQuery |
- Examples
 - Most of my work is proprietary and private; this selection of things I've managed to release publicly doesn't represent my full skillset.
 - 2020:
 - I'll be delivering a talk entitled [Set your data free with model-based architecture diagramming](#) at Write the Docs Portland in May
 - 2019:
 - Delivered a talk entitled [\(Architecture\) Diagrams as Data](#) at Clojure/conj ([video](#))
 - Presented [Data Driven Diagramming](#) at Clojure/nyc
 - Posted [Conference Highlights: Write the Docs Portland 2019](#)
 - 2018:
 - Presented [Concurrency via Communication — Large and Small](#) at the Bay Area Clojure Meetup (hosted by Funding Circle)
 - Released [the FC4 Framework](#), an open-source framework for authoring, publishing, and maintaining software architecture
 - Built [github-tools](#), a toolset for working with GitHub in a large org with many repos
 - Released [Learn You Some Kafka!](#) — atomic, bite-size, highly-focused interactive tutorials that teach how to work with Kafka using Ruby or Clojure
 - Presented [Large Nested JSON with Spec: A Comedy of Errors](#) at the Bay Area Clojure Meetup (hosted by Funding Circle)
 - Presented [Specifying Other People's Data Structures with Spec: an Experience Report](#) at Clojure/nyc
 - 2017:
 - Published [Kafka Summit 2017 Talk Proposal](#)
 - 2016:
 - Prototyped [Omnia](#), a system to search multiple document silos at once
 - Published [Strange Loop 2016 Recap](#) (blog post)
 - 2014:
 - Released [Three Things](#), a client-side Web app ([backstory involving Jeff Atwood](#))
 - Presented [The impedance mismatch of Web Microframeworks](#) at PyGrunn 2014 (hosted by Paylogic, a subsidiary of SFX)
 - Started collecting [resources for learning stream processing](#)
 - Designed [GitHub Digests](#), a system to aggregate GitHub notifications into daily email digests
 - 2013:
 - Published [Resourceful](#), a Clojure library that provides a more concise and more RESTful alternative to declaring routes.
 - My last major project for Arc90 was to build a new platform for reuters.com:
 - I created [a diagram](#) of the overall architecture of the “Reuters Next” platform
 - I was asked to review a draft of the API spec; I provided [these notes](#)
 - I developed [a document](#) to capture the culture and process of my team, so we could ramp new members up quickly and effectively
 - I determined that the API codebase I had inherited needed to be refactored, and wrote [this summary](#) to explain why
 - When I left the project I shared [my parting thoughts](#) with the team
 - I sometimes perform technical audits of existing applications or platforms. Here's [one example](#); I can provide more extensive examples upon request.

I'd be happy to provide more examples of my work, such as architecture diagrams and presentations, business analysis, code, etc. Please [get in touch!](#)

▼ Around the Web

- | | |
|-----------------------------|----------------------------------|
| • Blog | • Instapaper |
| • Twitter | • Stack Overflow |
| • Bookmarks | • GitHub |
- Hobbies
 - [Auto racing \(autox\)](#)
 - Cycling
 - Hiking
 - [Reading](#)
 - [Video games](#)

Contact Me

Thanks for scrolling all the way to the bottom of my résumé! I appreciate your interest. At this point, you might want to get in touch. Please do!

avi@aviflax.com

I'm based in [White Plains, New York](#) but prefer to work as part of a mostly or fully distributed team.

PDF Version

If you'd like to e-mail this résumé, or add it to your applicant tracking system, you're welcome to use [the PDF version](#).

Notes

1. Whereby “design” I mean “the creation of a plan or convention for the construction of an object, system or measurable human interaction” ([source](#)) — or, more succinctly, [“design is how it works”](#).

[Back to top](#)