Matt Kantor

https://mattkantor.com **Web**
the.matt.kantor@gmail.com **Email**
650 488 7405 **Phone**

Professional Summary

I'm a well-rounded software engineer with over two decades of experience.

I like working with distributed systems and anything involving concurrency. My career has mostly been web-adjacent, but I'm interested in other kinds of software too. I've been known to create tools and streamline workflows to help my colleagues be as productive and happy as possible.

Learning new programming languages is very fulfilling to me. Even if I don't end up using the language in my daily work, the experience teaches me to consider problems from different perspectives.

I use open source software and follow open standards whenever I can.

Specialties

- Full-stack development
- Distributed systems
- DevOps: observability, operations, CI/CD
- Developer tools
- User experience
- Domain modeling & API design
- Agile methodologies
- Big data

Programming Languages

- TypeScript
- Scala
- Rust
- Ruby
- Go
- Java
- PHP
- Objective-C

Tools & Services

- Docker & Kubernetes
- Google Cloud & AWS
- Databases: PostgreSQL, MySQL, MongoDB, Redis, BigQuery
- Data pipelines: Kafka, Dataflow/Beam, Google Pub/Sub
- Orchestration: Terraform, CloudFormation, Chef

Other Languages & Protocols

- HTML & CSS
- HTTP & REST
- Apache Thrift
- GraphQL
- SQL

Interests

I'm interested in most things IT, especially anything that has an impact on communication, society at large, or the ways we think and live. I'm also into physics, cosmology, ecology, sociology, philosophy, and just about anything that might help me make sense of this crazy and beautiful world. Designing things and solving problems are how I channel my creative energy to shape a better future.

Education

Bachelor of Arts in Computer Science from Ithaca College (Ithaca, NY) — May 2008

Employers

Clockworks, Inc. — Employee #1 (Software Engineer/Jack Of All Trades) — March 2021–August 2024
Credit Karma — Senior Software Engineer II — August 2017–June 2020
Kabam — Senior Software Engineer — May 2013–June 2017
Self-Employed — Software Engineer Contractor/Consultant — November 2011–March 2013
Sprongo.com — Software Engineer/Architect — February 2009–October 2011
Zampus, Inc. — Lead Developer/CTO — August 2008–August 2009

Selected Projects

Please
Passion Project — September 2024–Present

An experiment in programming language design.

Over the years I've accumulated various fringe ideas about programming language design, and decided to actualize them to see whether they can fit together. These ideas are mostly related to homoiconicity, "programmable" types (and perhaps dependent types), structural subtyping, abstract syntaxes which can be embedded in serialization/configuration languages, and "empirical types" to enable rich type safety without needing formal proofs.

I've been working on an open-source prototype implementation of Please (https://github.com/mkantor/please-lang-prototype).

Onalu
Clockworks — March 2021–August 2024

No-code workflow automation and collaboration with a reactive spreadsheet-like interface.

I joined Clockworks to work on Onalu as employee #1 after meeting the two co-founders and being impressed by their grit and vision for the company. It was clear to me that they had deeply thought about how to reduce incidental complexity in business processes, based on firsthand experience. Having been frustrated by this myself at times, their vision resonated with me.

Being the first employee meant I had to wear many hats. I contributed to everything from product design to business model development to implementation to customer support, and everything in-between.

Operator Web Server
Passion Project — June 2020–Present

A web server for small-to-medium sites.

Operator is meant for sites that are mostly content, but need a bit of backend logic/structure. It's designed to be flexible, making it easy to reuse existing assets/scripts and avoid lock-in. It serves static files directly, but it can also serve dynamic content that is generated at request time by handlebars templates or executables. Operator is written in Rust and uses the Actix web framework.

So far I've done all design and implementation myself. The project is open source (https://github.com/mkantor/operator).

Portals Product Platform
Credit Karma — January 2019–May 2020

A platform to empower product teams at Credit Karma to quickly build simple, personalized, cross-platform user experiences.

Credit Karma was experiencing growing pains: disjointed user experiences across product surfaces, features launching at different times on iOS/Android/web, and slow-moving/error-prone deployment processes. I founded an inter-organization task force whose goal was to overcome these problems. After an extensive planning and design process we came up with a set of complementary services and libraries comprising three main subsystems:

- A presentation layer which includes a cross-platform UI description language, allowing teams to produce native user interfaces for iOS/Android/web using a shared high-level definition.
- A data access layer to abstract data sources, surfacing user information as a collection of "facts" with a statically-typed schema.
- An orchestration layer for customization/targeting, and machinery to hydrate UI templates with dynamic data.

After building out the MVP, I was responsible for new capabilities like machine-learning-powered dynamic layout and styling of the UI, WYSIWYG authoring tools for non-technical creators, advanced performance metrics & analytics, and more. I eventually became tech lead on the project.

The platform continued to grow in both scope and scale, and by the end of 2020 it was used by most product surfaces in the app.

Selected Projects

"How To Scala" Class
Credit Karma — November 2017–April 2020

A bi-weekly class helping engineers become proficient Scala programmers.

I mentored more than 50 of my peers over the course of this program. My curriculum was a mix of theory and practice, with sessions usually introducing a concept then showing how to apply it with examples from real projects and live interactive coding sessions. I mostly covered intermediate-to-advanced topics, but presented them in ways that would break down mental roadblocks for beginners.

Topics included functional programming, concurrency/asynchronicity, approaches to automated testing, type theory (including algebraic data types and subtyping/variance), and basic category theory.

Bank Integration
Credit Karma — August 2017–January 2018

Infrastructure for Credit Karma to integrate with banks and debit accounts.

At Credit Karma there was a lot of discussion about how the product only understood "half of the balance sheet" (debts and lines of credit—from credit reports) and had little awareness of users' savings & cashflow. This made many desirable product features impossible.

Shortly after joining the company I was selected to co-lead a cross-departmental team to build a system for pulling user banking data and making it available to product services.

I architected and led the implementation of a collection of services to fetch/encrypt/persist bank account & transaction data, manage credentials, receive webhooks, and publish notifications about certain events. I also provided consultation for several product teams to help deliver user-facing features leveraging this data.

KASQ: Business Intelligence Pipeline
Kabam — August 2015–February 2017

A modern business intelligence pipeline leveraging Google Cloud Platform.

Kabam was a data-driven company and its business intelligence technology was a critical tool for decision-making. KASQ encompassed Kabam's entire data pipeline: streaming and batch-ingesting raw events, validating/transforming/aggregating those events, calculating business metrics, providing data to machine learning models, and pushing outputs to visualization tools and third-party APIs. KASQ was composed of many cooperating sub-components to handle these responsibilities; most were written in Scala and made use of Google Cloud services such as BigQuery, Dataflow, and hosted Kubernetes clusters. By the end of 2016 KASQ was processing 10 million events per day, with over 300TB of data stored in BigQuery.

I worked on many different pieces of KASQ, taking the lead on major features like data ingestion and metrics to support user acquisition, enhancing concurrency across large swaths of the pipeline to improve throughput, and a monitoring framework to keep track of the system's health.

Selected Projects

Self-Healing MongoDB Cluster

Kabam — September 2013–January 2014

Automated infrastructure for a sharded MongoDB cluster on AWS.

I designed and built infrastructure-as-code for a MongoDB cluster with three config servers, an administrative node running `mongos`, and a configurable number of shards, each including a hidden secondary responsible for taking snapshots.

The cluster was designed with a "cattle, not pets" mentality and was robust to failures. As long as each replica set had at least one live member and there was at least one operable config server, the rest of the cluster could go down without impacting consistency or availability. All failover was completely automated, with replacement nodes loading state from snapshots in order to prevent a costly full sync. Tools used included Chef, AWS CloudFormation, and AWS CloudWatch.

SkiClubZ/Sprongo

Sprongo.com — February 2009–October 2011

A web-based platform for amateur & professional athletes to improve their performance using video analysis and remote coaching.

I helped develop these projects from their beginnings as part of a very small startup group. SkiClubZ.com was our proof-of-concept, focused on the skiing and snowboarding markets. After a lot of iteration and user feedback we had developed a solid set of tools and a way of packaging them into a helpful, easy-to-use service. We refreshed the UI and simplified the feature set to create Sprongo, a cross-sport solution based on the learnings from SkiClubZ.com.

The site's focal point was an HTML5-based video tool with variable rate slow-motion, video overlays, split screen, frame-stepping, drawing, and timestamped in-video comments. Because the HTML5 video API was still young when we developed these features, there were lots of workarounds for browser inconsistencies in order to deliver the best experience possible to each of our users.

Since the company was so small I did a bit of everything, but focused on development, project management, operations, interaction design, testing/QA, and user research.