



David Ellis

Location: San Antonio, TX **Phone:** 210-370-7888 **Email:** david@conquerthelawn.com

Github: github.com/davidkellis

01 SUMMARY

I'm a maker. Building software to solve practical problems is my craft. Most recently I've been incorporating LLMs and agent-assisted development into my development process. Agent-assisted engineering has turned projects that might have been an entire career into something I can achieve in six months. LLMs are such a game changer because they let me focus my attention on executing a vision for the product, subject to design constraints that I define, rather than spending my time implementing those constraints by hand.

I'm experienced

- 20 years of professional software development experience.
- Design patterns, integration patterns, data structures, algorithms, and domain modeling are how I think.
- Led and implemented numerous projects and product features from concept to maintenance.
- I've done backend, frontend, infrastructure, and everything in between.

I'm smart

- I communicate clearly and write well.
- I can identify problems in the business domain and solve them.
- I'm practically and analytically minded.
- I learn quickly.

I get things done

- I'm pragmatic and productive.
- I work on personal side projects to learn and build experience with new technologies, tools, and techniques.
- I take ownership in the product suite my team owns.

I'm pleasant to work with

- I'm a team player.
- I'm not a know-it-all.
- I'm quick to acknowledge and address my mistakes.
- I'm objective and reasonable.

02 WORK EXPERIENCE

Senior Software Engineer

Loop Returns

📅 06/2023 – 10/2026 📍 Remote, US

Self-serve Ops

- Built self-serve tooling enabling application engineers to package and deploy Loop's PHP monolith across multiple Kubernetes environments without requiring deployment assistance from Operations; increased release frequency from 1-2 releases per day to 5-12 per day.
- Built self-serve tooling enabling application engineers to change application runtime configuration and secrets without the need to request assistance from Operations; staging and production config changes went from days to minutes.

Track by Loop Improvements

- Contributed to integrating Wonderment's shipment-tracking platform (acquired and rebranded as Track by Loop) into Loop's application suite, enabling merchants to provide shipment tracking and notifications to customers.
- Extended the Track by Loop shipment-tracking platform to be merchant-platform agnostic rather than Shopify specific, enabling Loop's sales team to pursue Salesforce Commerce Cloud merchants.
- Cut the time to report on a merchant's shipments from minutes to milliseconds.

Senior Software Engineer**Shopify**

📅 05/2022 – 05/2023 📍 Remote, US and Canada

Protected Customer Data (PCD) Review Tooling

- Designed and built internal privilege review tooling to enable the App Store App Review team to audit, review, and adjust the access privileges to protected customer data that App Store apps have; enabled Shopify to restrict app access to customer PII.

Shopify OAuth 2 Implementation Redesign

- Contributed to a redesign of Shopify's OAuth 2 implementation such that apps without a need for Shopify API privileges could be installed into a merchant's online shop without participating in the OAuth 2 authorization code handshake; dramatically simplified the development of Shopify App Store apps, and opened up the App Store to an entirely new category of application: serverless apps.

Senior Software Engineer**PagerDuty**

📅 04/2019 – 04/2022 📍 Remote, US and Canada

High Throughput Rewrite

- My team and I rewrote PagerDuty's event ingestion pipeline within three years in order to sustain an event ingestion growth rate of 60% annually.
- New ingestion pipeline architecture gave us horizontal scalability that we had previously lacked.

Event Buffer Application

- My team and I reduced the frequency that PagerDuty's event ingestion API throttled inbound "burst traffic" by 70%.
- By buffering three minutes worth of customer event submissions, we removed our customers' burden of evenly distributing event flow in order to avoid being throttled.

Junior -> Senior Software Engineer**Rackspace**

📅 09/2010 – 03/2019 📍 San Antonio, TX

Bridge Service

- Designed and built a VLAN bridging tool that connects layer 2 networks (VLANs) from multiple data centers together so that devices on each distinct network (VLAN) can talk as if they were on the same local network (VLAN).
- The VLAN bridging tool formed the foundation of the RackConnect 3.0 product, which enabled customers to expand their footprint across multiple data centers while logically residing within the same layer 2 network.

NAT Microservice

- Led a project to extract the management of NAT (network address translation) translations out of a legacy Ruby on Rails backend into a new NAT microservice written in Golang.

IP Space Microservice

- Co-led a project to extract the management of the IP space owned by Rackspace out of a legacy Ruby on Rails backend into a new IP management microservice written in Golang.

Research and Teaching Assistant**Texas Tech Computer Science Department**

📅 01/2009 – 08/2010 📍 Lubbock, TX

- Ported NASA-supplied MATLAB programs into SequenceL for verification and automatic parallelization.
- Conducted programming labs, created lab assignments, and graded class assignments for CS 1411 (Intro. to Programming with Python) and CS 2413 (Data Structures).

Java Developer**Anderson Merchandisers**

📅 05/2006 – 08/2008 📍 Amarillo, TX

- Extended Anderson's Java shipping software to ship out of multiple automated distribution centers instead of just one manual-ship DC, resulting in more than 50x shipping capacity improvement. Improved capacity gave Anderson the ability to ship walmart.com orders for the launch of the 7th Harry Potter book.
- Added support for routing incoming orders to third-party vendors into Anderson's Walmart.com order processing system, enabling Anderson to re-route manufacture-on-demand DVD orders to Hewlett Packard.

03 SKILLS AND PRACTICE

With a modern LLM and agent harness I can build nearly anything in any tech stack.

Prior to modern LLMs, I could build nearly anything involving any of the following:

Languages: Ruby, Crystal, Go, TypeScript, JavaScript, Scala, Elixir, F#, Java, C#, Python, PHP, etc.

Data stores: PostgreSQL, MySQL/MariaDB, Snowflake, Cassandra, MongoDB, DynamoDB, RDS, Redis, etc.

Messaging: Kafka, SQS, RabbitMQ/AMQP, ActiveMQ, NATS, NSQ, etc.

Whether AI-assisted or by hand, I generally try to nudge codebases toward following the SOLID design principles, a layered/tiered architecture, a clear domain model with well-defined semantics, sufficient unit/integration/e2e testing, and being as operationally simple as possible.

04 EDUCATION

M.S. in Computer Science

Texas Tech University

📅 08/2008 – 08/2010 📍 Lubbock, TX

B.S. in Computer Science

West Texas A&M University

📅 08/2002 – 05/2006 📍 Canyon, TX

05 PERSONAL PROJECTS

Arborist

🌐 github.com/davidkellis/arborist

A PEG parser combinator that supports left and right recursion, written in Crystal.

Able

🌐 github.com/davidkellis/able

Able is a new programming language that takes cues from Ruby, Scala, F#, and Go. Using LLMs and agentic engineering I have built a tree-sitter parser, two tree-walker interpreters (typescript and golang), a bytecode interpreter, a golang transpiler, and a compiler to JVM bytecode.

SecuritiesDB

🌐 github.com/davidkellis/securitiesdb

A Ruby implementation of a securities master database.

simstring

🌐 github.com/davidkellis/simstring

A Ruby implementation of the SimString approximate string matching algorithm.

pls

🌐 github.com/davidkellis/pls

A Scala implementation of the partial least squares (PLS) algorithm. PLS is a relatively unknown yet very robust linear regression algorithm that can identify linear and even curvilinear relationships between predictor and response variables.

fsm

🌐 github.com/davidkellis/fsm

A finite state machine combinator library for constructing, via a DSL, recognizers for regular languages.

tradesim_scala

🌐 github.com/davidkellis/tradesim_scala

A backtesting tool I built to test and compare trading strategies.

stocktrader_clojure

🌐 github.com/davidkellis/stocktrader_clojure

A Clojure implementation of a stock trading strategy backtesting tool I wrote for my master's thesis.