

Andrew Willette

willette.andrew1846@gmail.com

815 218 4802

Personal Website: andrewwillette.com

Github: github.com/andrewwillette

Linkedin: linkedin.com/in/andrew-willette-28792811a/

I am a Software Engineer who's built production-running software since 2017. My main technology stacks are around services running in the cloud. This includes frontend and backend services written in Go, Java, C#, and Node. I am passionate about building and delivering clean, well-tested, and easy-to-maintain software that satisfies business needs.

Skills

AWS go neovim docker Splunk Terraform New Relic lua rust java bash zsh C scala
typescript Packer oracleDB css sqlite Cassandra ubuntu RHEL Spring markdown metasploit
music theory

Experience

PWC - Engineering Director 2025 - Present

Discover Card Acquisition Integration with Capital One

- Directed contracts integrating Discover Card acquisition flows onto Capital One platforms
- Owned personnel across 5 teams and served as technical lead for 1 team
- Partnered across a large engineering organization to deliver a wide variety of cloud-hosted services
- Drove delivery of frontend and backend components, including end to end integration testing

Kunai - Engineering Director 2024 - 2025

- Directed technical leadership for 10 teams designing and implementing cloud software solutions for Capital One
 - Teams consisted of 1 technical lead, 5 Senior Software Engineers, and 1 Product Manager
- **Educated Kunai's teams on the client's cloud architecture**, streamlining use of the client's systems
- Ran scrum ceremonies, turning client-provided initiatives into a JIRA backlog
- Conducted **100+ technical and behavioral interviews** to help build a program based in Mexico City, Mexico
- Key contact with Capital One's technical leadership, keeping delivery aligned with expectations

Banking Core Consolidation

- Directed a legacy-to-modern core data migration, including real-time/batch Java+Spark pipelines and a Go Lambda ingestion service
- Directed adoption of the new Modern Banking Core across client applications (Java, Go, TypeScript, Python), including integration/load testing on its GraphQL APIs
- Directed a self-service data-comparison tool migrating Databricks workflows to AWS Glue

Kunai - Senior Software Engineer 2022 - 2023

- Design and build microservices `Participants-Datastore`, `Collaterals-Datastore`, and `Account-Attribute-Datastore` for managing various properties of Auto Loan accounts managed by Capital One.
 - Implemented persistence layers using a Cassandra database.
 - Managed a multi-region AWS Fargate deployment strategy for individual services.
 - Managed a docker-compose local deployment strategy for standing up a local Cassandra database.
 - Implemented advanced mocking patterns using testify/mock package.
- Design and build an `Account-Maintenance` microservice architecture for tracking and making available changes to

an Auto Loan account.

- Implemented an avro schema for managing dynamic attributes reported from Capital One's underlying "account ledger" allowing for runtime-updates to which attributes can be tracked as "changed".
- Implemented automated pipeline for updating the avro schema in AWS S3.
- Implemented a go module "client library" imported into several Capital One microservices to facilitate reporting updated values to the `Account-Maintenance` service.
- Integrated New Relic into go microservices.
 - apm-agent for managing AWS infra behavior.
 - `zipkin` tracing protocol for tracking call-stack timing and behavior.
 - Distributed tracing to track requests across microservices.
- Manage the deployment of services into a multi-pre-prod and prod environment ecosystem.

Fiddling Technologies LLC - President : 2017 - 2023

- Personal website built on purely go stack.
- Manage and play violin around my local area, primarily restaurants and outdoor markets.

Cerner - Software Engineer II - Identity and Access Management : 2017 - 2022

- Built a React Single Page Application(SPA), currently hosted at login.cerner.com, which manages Cerner users from any client(and consequentially identity provider) through a login workflow which integrates them into Cerner's Single-Sign-On ecosystem.
 - Frontends built using Cerner's open-source `terra` React components.
 - Redux "backend" for the SPA frontend.
 - Wrote client javascript code for consuming Cloud IAM REST API's for managing the login workflow.
 - Wrote client javascript which consumed GraphQL endpoint, allowed for large, flexible use cases related to selecting a client, tenant, and application.
 - The result of a valid application selection is a redirect to the proper identity provider, where a valid login sends the user to their desired client's environment, and triggers persistence in Cerner's Session Management cloud solution.
- Added support in enterprise User Roster, a solution for representing users from all identity providers in one Cerner-managed cloud location, for inactive users. Allows for persistence of old account resources rather than deleting them.
 - Passively updated oracleDB schema to include active column.
 - Updated authorization model of existing OAuth web services to insure inactive accounts could not utilize services.
 - Updated front end User Interface for managing enterprise users to allow for displaying inactive users in their own respective table. This involved JSP frontends and custom CSS.
- Updated Cerner's solution for managing OAuth 1.0a accounts to allow for an authentication workflow utilizing [JSON Web Tokens](#).
 - Updated the account UI page to allow for registration of a JSON Web Key Set, this is a public-key representation of a asymmetric cryptography implementation, where the respective private key is used to sign JSON Web Tokens, allowing for authentication.
 - Wrote the Spring Controller handling jwt signature assertions, utilizing the open-source [nimbus-jose-jwt](#) library.
- Updated Cerner's solution for managing [OAuth 1.0a](#) accounts/credentials to allow for automatic(non-human vetted) creation of lower-privileged accounts via REST APIs.
 - Wrote Java Spring Rest API Controllers for managing/validating requests to create new lower-privileged accounts.
 - Passively updated oracle database schema to allow for more extensive authorization model with lower-privileged accounts.
 - Updated the front-end management User Interfaces for the accounts to display them properly with lower-privileges. This involved pages generated with the JSP java technology and custom css.
- Manage the development and release of 5 different cloud-hosted solutions.
 - Each solution has a [Packer](#) and [Terraform](#) configuration for AWS representation.
 - Amazon Machine Images(AMIs) are created and managed via Chef Infrastructure.
 - Java artifacts are released into corporate Maven repository.
 - npm artifacts are released into corporate npm repository.

- Continuous deployment of new AMIs into AWS are managed via a jenkins instance. I have extensive experience debugging issues in this environment.
- Uplifted Cloud IAM's jenkins solution to be hosted in AWS.
 - Master and slave nodes have custom Packer and Terraform configurations for representation in AWS
 - The logic for creating and managing jenkins jobs was accomplished via jenkins native support for [groovy scripting](#) and aws client libraries.
 - Wrote groovy shutdown hooks to unmount jenkins master EBS volumes prior to shutting down the instance, allowing for persistence of jenkins jobs data.
- Implemented a client-side Typescript library for managing a user's authentication state in browser Single Page Applications.
 - Consumed by clients which rely on Cerner's [single sign on](#) ecosystem, with the library's APIs exposed via npm package import.
 - Allows consumers to configure their applications with particular identity providers in Cerner's ecosystem, wherein the client library triggers logins, session locks and other session management utilities via REST requests, with client cookies, to Cerner's Session Management and Authorization Management web services.
- Implemented a variety of logging frameworks and patterns.
 - Install, configure and use logs from [new relics java agent](#).
 - Install, configure, and use logs from Splunk. Used Cerner's open source solution [splunk-pickaxe](#) to write custom Splunk alerts as code. The resultant alerts are emailed to our Scrum's Teams channel.
 - I make thoughtful decisions when configuring java's logback framework to allow for effective troubleshooting in Splunk.

GOM Software International - Software Engineer : Agricultural Feeding, 2018 - 2020

- Designed and implemented complex matrix in tables in [knockout.js](#) to update feeding information for livestock.
- Implemented Braintree online payment system using the [Braintree .net SDK](#), allowing for collecting credit and debit cards for a scheduled payment processing plan.
- Extended Microsoft SQL Server schema for managing livestock feeding data, such as "Dry Matter Percentage" present in food fed to cattle.
- Implemented Dotnet's [role-based authentication](#).

Education

- University of Iowa, 2014 - 2018
 - Bachelor's Degree: Computer Science
 - Bachelor's Degree: Violin Performance