Abhi Rao ¦ Houston, TX ¦ contact@abhirao.dev

Skills

Technologies

- Ruby
- JavaScript
- Git
- Rails
- MongoDB
- Python
- PostGreSQL
- TypeScript
- React

Experience

Planet FWD — June 2023 - Present

Engineering Lead

Overview

Planet FWD is a startup in the ClimateTech space. It offers a platform to create Life-Cycle Assessments (LCAs) at the product level as well as Corporate Inventories to measure the impact of companies according to the GHG protocol. Joined this startup as it was trying to change from a software-enabled service to a SaaS.

Responsibilities

- Technical leadership
- Epic Breakdowns
- Process Improvements

Projects

Initial Buildout

Led the team through the initial buildout of the SaaS offering. Adjusted architecture to allow entire team to contribute across the tech stack. Personally contributed across the entire stack: PostGres DB design, .NET API layer, Python calculation engine and React front-end.

Leveraging AI for features

Replaced hard to maintain custom recommendation engine with a search implementation based on embeddings. Added features that would let users get started with minimal input by having AI-provided default values. Guiding principle: AI to be used to help users get their data in the software and all AI-provided data to be marked as such.

Scaling up

Optimized flow for core emission calculations by identifying bottlenecks which unlocked autoscaling.

Onit, Inc. May 2012 - June 2023

Senior Software Architect

Overview

Onit is a provider of workflow solutions in the LegalTech space. Joined the startup as one of the first dozen employees. Key contributor to the platform that grew ARR from <1M to >40+M. Provided technical leadership and mentorship to new developers. Worked closely with Product Management to understand how features can be added to a complex platform without impacting existing customers.

Responsibilities

- Technical leadership
- Merge reviews
- Security Evaluations
- Performance Improvements

Projects

Initial Platform Buildout

Worked on the initial buildout of the workflow platform (using Ruby on Rails) that would be the basis of Onit's product offerings. The platform needed to be powerful enough to solve any problem involving approval and collaboration but unopinionated so that any customer's business process could be modeled with configuration but no code change.

Data migration from MongoDB to PostGreSQL

While the schema-less approach of MongoDB made it easy to get started, the lack of relationships and cross-document transactions were a severe limitation. Designed a migration process where data from MongoDB could be moved to PostGreSQL seamlessly. The data scaled up to millions of rows and had the ability to migrate customers with minimal downtime (minutes, not hours).

Front-end Rewrite using React

The level of complexity of the application led to many performance issues and bugs in the prior UI framework. Championed React as the framework for a major UI rewrite in 2016. React's declarative model and one way data flow led to a more stable application and better isolation of responsibilities.

J.P. Morgan Chase May 2006 - May 2012

Senior Developer

Overview

Joined the firm as an intern during final year in university. Quickly became an integral part of the team. Learned all parts of application stack and business domain of application. Became technical lead for the team after two years.

Responsibilities

- Implemented changes from front to back end. Worked on Web, UI, Batch applications, DB changes and APIs
- Designed changes to address business needs and non-functional requirements
- Responsible for management of SVN repo and maven build

Projects

Approvals Recording and Control

ARC was a J2EE web application with 4000+ internal users across the globe. It was used across LOBs to grade clients and approve credit facilities and transactions. All proposed changes would go through a workflow where required approvals were calculated and recorded. It utilized EJBs and MDBs and had interfaces with several applications. It used Spring, Hibernate and Struts.

Introduction of Letters of Credit to ARC

A new workflow was added to allow users to approve Letters of Credit via ARC. LC requests would come in through a new MQ interface with a mainframe application based in Hong Kong. Worked with architect to design application changes. Responsible for integration between Websphere MQ and Weblogic. Designed and implemented a fixed-length message parser. Other changes included workflow, UI and timed jobs for notifications and automated rejections of eligible LCs.

Reengineering of ARC

To improve the application performance as experienced by users ARC was split into two - foreground and background. All UI interaction was encapsulated in the foreground app. Any long running tasks which could be performed asynchronously were moved to a new background application. Responsible for moving various workflows from a UI based synchronous model to asynchronous background tasks. Refactored so that foreground and background components did not duplicate any code.

Onboarding Asia/EMEA regions for Letters of Credit Approval

Two years after initial LC implementation, Asia and EMEA regions needed to be onboarded. These regions required some changes to the workflow but the challenge was to reuse the code base and triple the load without impacting the user experience. Designed application changes and directed the offshore team that implemented the changes. Designed region handling so that addition of new processing units would require some database configuration but no new application code.

Education

University of Houston 2002-2006

- B.S., Computer Science. Minor in Mathematics and Philosophy