

# SOFTWARE DEVELOPMENT PROJECTS

*Working in an agile environment (TDD, BDD, DDD Pair Programming, etc.)*

This provides a clear overview of some of the significant contributions I've made at each company.

All the images displayed are publicly accessible to anyone who has purchased these products and are included here solely for illustrative purposes.

## Core Pillars / Philosophy

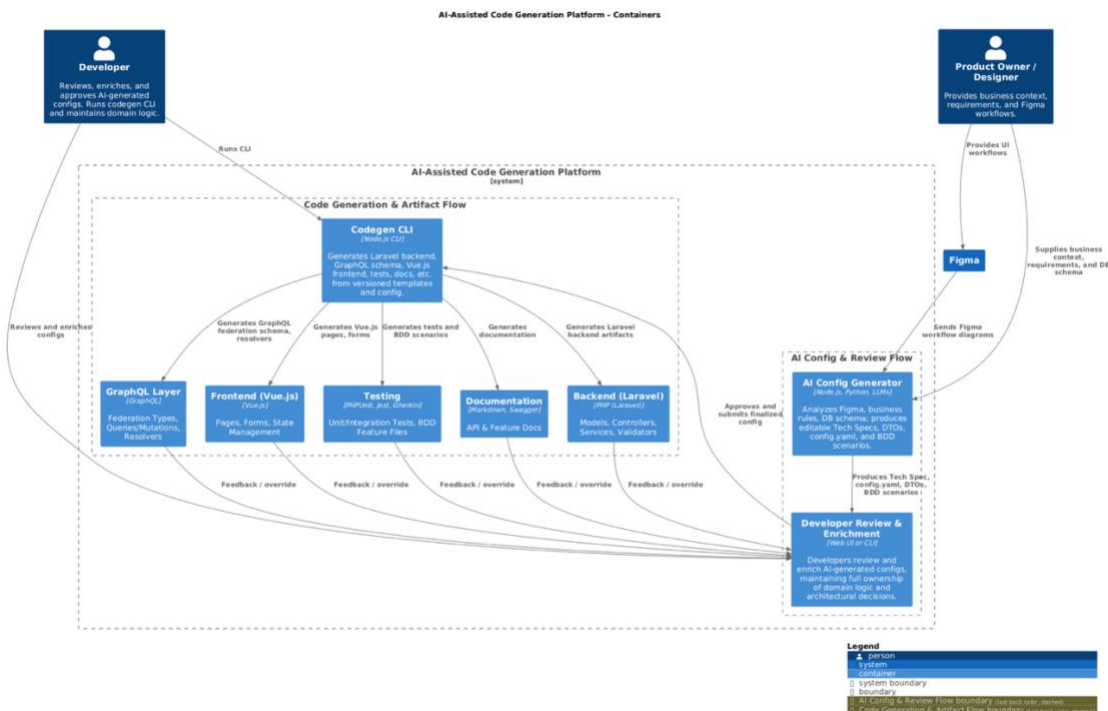
---

- ◇ Intuitive & Dynamic Design
- ◇ Hide Complexities away from Developer
- ◇ Promote Developer Happiness
- ◇ Accessible Architectural Design
- ◇ Cross Team Collaboration

# Helcim

Lead SENIOR SOFTWARE DEVELOPER/ARCHITECT

## AI-ASSISTED CODE GENERATION PIPELINE



### Problem:

Helcim's **custom PHP web application** enabled years of business growth, but the **legacy architecture** has become a bottleneck for **scalability** and **feature delivery**. The organization prioritized migration to a **Laravel + Vue.js stack** and the **ethical adoption of AI-driven engineering automation**—ensuring **developers remain in control**—to accelerate **development velocity** and support **sustainable growth**.

## **Solution:**

Built a **schema-driven, API contract-first pipeline** where **AI guides developers** to create config files from Figma flows, DB schemas, and business rules. These configs drive the **codegen CLI**, generating production-ready, fully tested features—keeping **developers in control** and eliminating boilerplate.

## **Outcomes:**

- **Faster feature delivery** — from weeks to hours.
- Consistent, contract-first architecture on every feature.
- **Developers in control** — focus on logic, not boilerplate.
- Easy onboarding via schema-driven, editable configs.
- Safe, iterative improvements with versioned templates.
- **CI/CD ready** — all code is test-covered and production-ready.
- Sustained competitive edge through scalable automation.

# MAJOR CLARITY BY PAPER

---

Lead SENIOR SOFTWARE DEVELOPER/ARCHITECT

## EFFICIENT LISTING IMPORTER AUTOMATION

**Problem:** There was a need to integrate the acquired MajorClarity web application into PAPER's ecosystem, ensuring it operated seamlessly and transparently to the user within PAPER's platform.

This migration needed to resolve fragmented user authentication, implement a unified Single Sign-On (SSO) solution, and ensure consistent rostering and student data management across both platforms.

**Solution:** I led the implementation of Auth0 SSO via Google, Clever, and Classlink providers, unifying authentication and improving security. I devised workflow-oriented rostering and student data sync service, ensuring consistency across both platforms.

### Outcomes:

- Unified authentication with Auth0 SSO, enhancing security and user experience.
- Ensured consistent rostering and data synchronisation across platforms via RBAC.
- Improved scalability and modularity with SPA and MFE architecture using VueJS.
- Modernised the Rails monolith for greater development efficiency.

# SPACELIST

---

SENIOR SOFTWARE DEVELOPER/ARCHITECT

## EFFICIENT LISTING IMPORTER AUTOMATION

**Problem:** Spacelist faced inefficiencies in handling high volumes of property listings, prolonged onboarding times, and significant processing costs. These issues led to processing delays, frequent file preview errors, and increased operational costs due to the need for manual interventions.

**Solution:** We built an automated listing importer process with a declarative configuration per client, including field mapping, reducing onboarding time from weeks to days. We also significantly cut redundant asset re-upload costs by detecting duplicates based on E-Tags and filenames.

### Outcomes:

- Onboarding new clients were reduced from weeks to days.
- Sitemap generation time was cut from 6 hours to 50 minutes.
- Resolved file preview errors by bulk updating S3 assets with accurate metadata.
- Reduced costs by skipping redundant image uploads.

# GEOFORCE

---

SENIOR SOFTWARE DEVELOPER/ARCHITECT

## OKTA INTEGRATION AUTOMATION

**Problem:** Geoforce faced challenges with a fragmented user management system that required multiple login credentials across various applications. This has resulted in security vulnerabilities, a poor user experience, and increased operational costs due to the frequent need for password resets and access management.

**Solution:** Okta was introduced as the central SSO solution to unify user authentication. We built a custom-branded Okta login service that brokers authentication across all applications and includes migrating pre-existing users to Okta.

We also introduced Infrastructure-as-Code using Okta Terraform plugins to automate deploying Okta resources (apps, user profiles, groups, and security policies).

### Outcomes:

- Unified user authentication across all applications, improving security and eliminating the need for multiple logins.
- Custom-branded Okta login service provided a seamless and consistent user experience.
- Pre-existing users were migrated to Okta with minimal disruption.
- Okta Terraform plugins automated the deployment of resources, reducing manual effort.
- Scalability and security were improved through automated provisioning and consistent enforcement of security policies.

# KICKBOOSTER

---

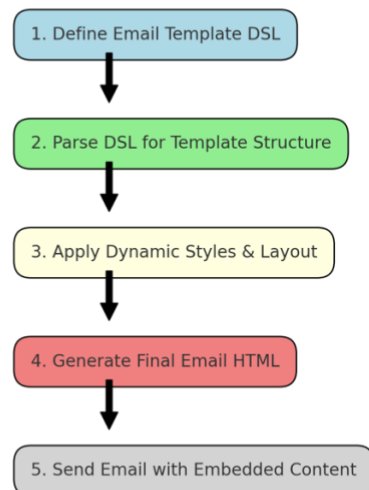
SENIOR SOFTWARE DEVELOPER/ARCHITECT

## DYNAMIC EMAIL TEMPLATE BUILDER

**Problem:** The manual and inconsistent creation of client emails led to frequent support calls due to broken or invalid links, significantly increasing maintenance costs and hindering operational efficiency.

**Solution:** Built a reusable email template system using a lightweight DSL. This enabled consistent styling, seamless deployment, and non-technical content adjustments. Automated validation ensured consistent content and functional links, reducing support issues and improving reliability.

Dynamic Email Template Builder Workflow



# RELAY PLATFORM

---

SENIOR SOFTWARE LEAD DEVELOPER/ARCHITECT

## CYBER INSURANCE QUOTING PLATFORM

**Problem:** Introduce a standalone multiple-carrier Quoting Platform designed to simplify the generation of quotes by tailoring them to customer requirements, terms, and endorsements.

This project stands as one of the most ambitious endeavours I've had the privilege to be a part of. I take great pride in it, primarily because of the outstanding teamwork, collaboration, and mentorship that have been integral to every facet of the project's success.

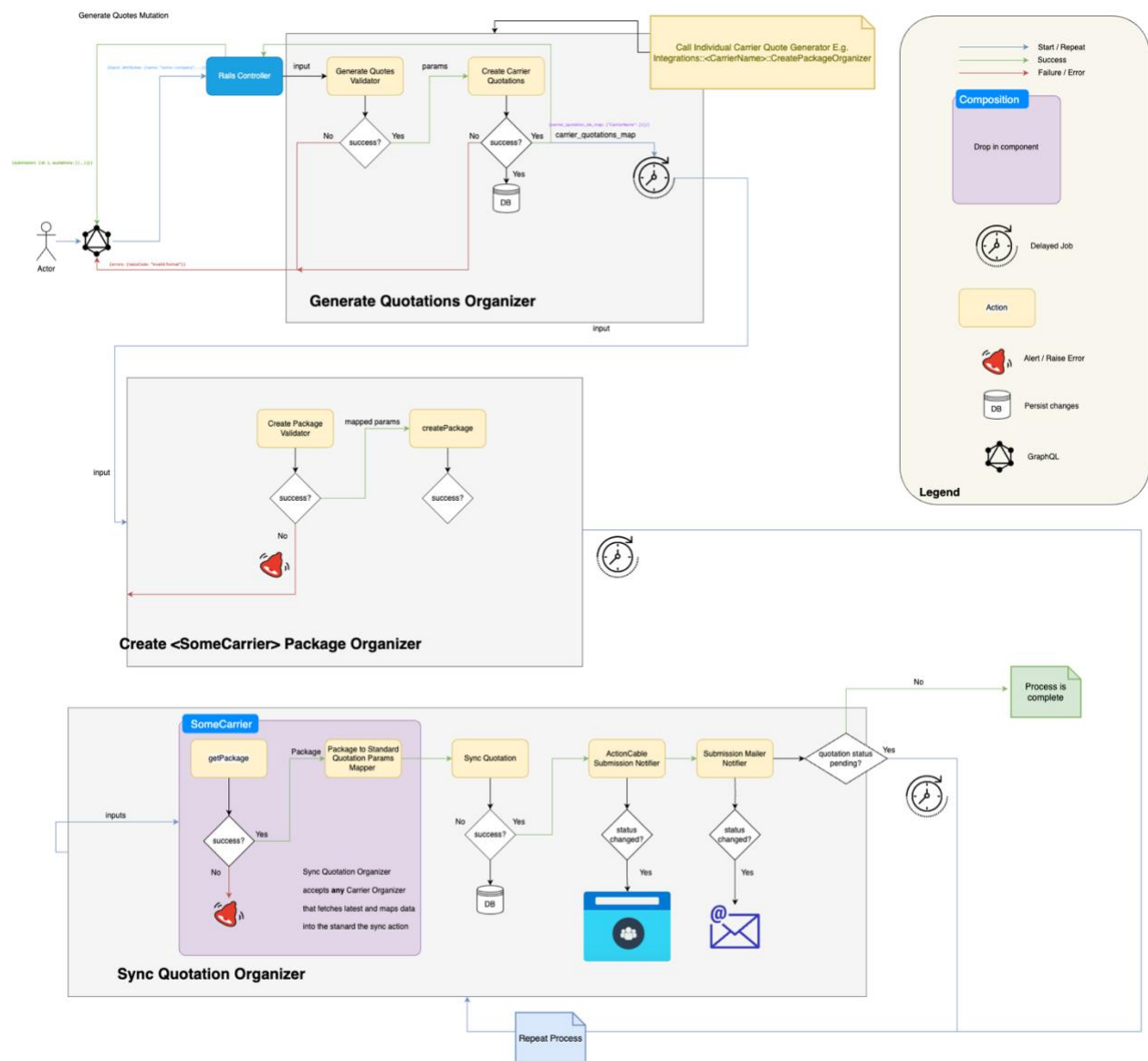
A particularly distinctive feature of this project, which played a crucial role in shaping its architectural design, was the requirement to onboard multiple broker carriers. These carriers each presented unique API workflows, encompassing the entire process from generating initial Quotes to completing renewals.



## Workflow-Oriented Async Services

We purposefully established a flexible architecture during the integration of external carrier Quoting APIs, responsible for transforming quotes into a standardized format.

This approach enables our core workflows to be applicable to all quotes, whether they originate from carriers or are manually generated. For instance, when a quote transitions from a pending state to a generated state, a series of general tasks, such as email notifications and webhook broadcasts, are triggered. These tasks are executed with full support for asynchronous operations and automatic refresh to guarantee a smooth user experience.



## Dynamic Components

Given the extensive permutations involved in the Cyber Insurance Quoting Application, which encompasses fundamental proposal information, underwriting, endorsements, and conditionals, it became essential to develop exceptionally dynamic components driven by stateless configurations.

This scalable approach has made the Quoting Platform the most successful product in Relay Platform's history and has been designed to grow in tandem with the business.

## JSONSchema Driven Dynamic Components

Return to Dashboard

Place

Analyze

Manage

Integrate

Request a Feature

Your Logo Here

New Cyber Application

Company Name

Software Inc.

Contact Email

https://www.swinc.com

Client Name

Jill Erickson

Client Email

jperickson@mainmotor.com

Street

275 First Ave

City

New York

State

New York

Zip Code

10001

Phone Number

+1 (834) 153-4927

Company Description

423430 - Computer and Computer Peripheral Equipment and Software Merchant Wholesalers

Underwriting Code

Computer Software, and Electronics Wholesalers (Non Industrial Only)

Number of Employees

65

Number of Records

None

Annual Revenue

\$14,000,000

Limit

\$2,000,000

Deductible

\$5,000

Policy Effective Date

Sep 14, 2021

PREFERRED CARRIERS (5)

Carrier A

Carrier B

Carrier C

Carrier D

Carrier E

UNDERWRITING QUESTIONS

SHOW UNDERWRITING QUESTIONS

ADDITIONAL CARRIERS (1)

OR

Add New Carrier

Allstate North American Insurance Company

Sarah Allstate

Your Email To Carriers

Normal

Best Regards,

Jill Erickson

Supporting Documents

Upload Document

I agree to share placement data over email and via Relay with the invited insurers. See Terms and Conditions and Privacy Policy.

I agree with the Carrier's statements of fact.

Save Draft

Get Quote

Chat

Help

© 2021 Relay Platform Inc.

Terms and Conditions

Service Level Details

Privacy Policy

Return to Dashboard

Edit Application

View Application (PDF)

All Documents

Cyber Quote for ABC Insurance Co

Limit: \$500,000

Deductible: \$5,000

Get New Quotes

Add Quote Manually

Export All

Showing 5 of 5 quotes

Filter by limit

Filter by deductible

Carrier A

Carrier B

Carrier C

	\$500,000   \$5,000	\$500,000   \$5,000	\$500,000   \$5,000
Broker Recommended	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Broker comment	Lower deductible	A++ rated carrier	Lowest premium
Limit	\$500,000	\$500,000	\$500,000
Deductible	\$2,500	\$5,000	\$5,000
AM Best Rating	A+	A++	A
Admitted / Non-Admitted	Admitted	Admitted	Non-Admitted
Premium	\$510.26	\$1,073	\$1,010
Fee	\$0	\$0	\$195
Documents	# ABC Insurance...pdf	# ABC Insurance...pdf	# ABC Insurance...pdf
View Details			
Bind Quote	Bind Quotes	Bind Quotes	Bind Quotes
Underwriter	General Inbox 1 hour, day response	Default Underwriter 1 hour, day response	Travis Landrum 1 hour, day response

ENDORSEMENTS

\$500,000 | \$2,500

\$500,000 | \$5,000

\$500,000 | \$5,000

	\$500,000   \$2,500	\$500,000   \$5,000	\$500,000   \$5,000
Deductible	24-hour retention waiver	24-hour retention waiver	24-hour retention waiver
Full Retroactive Coverage	Yes	Yes	Yes
Actions of Rogue Employee	Rogue Employee	Rogue Employee	Rogue Employee
Risk Management Services	Risk Score/Training	Risk Score/Training	Risk Score/Training
Third Party	Limit: Deduct/Waiting	Limit: Deduct/Waiting	Limit: Deduct/Waiting
Media Liability	\$500,000 \$2,500	\$500,000 \$5,000	\$500,000 \$5,000
Network Security & Privacy Liability	\$500,000 \$2,500	\$500,000 \$5,000	\$500,000 \$5,000
Payment Card Loss	\$500,000 \$2,500	\$500,000 \$5,000	\$500,000 \$5,000
Regulatory Proceedings	\$500,000 \$2,500	\$500,000 \$5,000	\$500,000 \$5,000
Technology Errors & Omissions	-	-	-
TCRA Defense Cost Coverage	-	-	\$50,000
First Party	Limit: Deduct/Waiting	Limit: Deduct/Waiting	Limit: Deduct/Waiting
Cyber Incident Response Team	\$500,000 \$2,500	\$500,000 \$5,000	\$500,000 \$5,000
Business Interruption Loss	\$500,000 \$2,500/8 hours	\$500,000 \$5,000/12 hours	\$500,000 \$2,500/8 hours
Contingent Business Interruption Loss	\$500,000 \$2,500/8 hours	\$500,000 \$5,000/18 hours	\$500,000 \$2,500/8 hours
Business Interruption - System Failure	\$500,000 \$2,500	-	\$500,000 \$5,000/8 hours
Contingent Business Interruption Loss - System Failure	\$500,000 \$2,500	-	\$500,000 \$5,000/8 hours
Reputational Harm	\$500,000 \$2,500/14 days	-	\$500,000 \$5,000/2 weeks
Digital Data Recovery	\$500,000 \$2,500	\$500,000 \$5,000	\$500,000 \$5,000
Network Extortion	\$500,000 \$2,500	\$500,000 \$5,000	\$500,000 \$5,000
Cyber Crime	Limit: Deduct/Waiting	Limit: Deduct/Waiting	Limit: Deduct/Waiting
Computer Fraud	-	\$100,000 \$5,000	-
Funds Transfer Fraud	\$125,000 \$12,500	\$100,000 \$5,000	\$250,000 \$5,000
Vendor or Client Payment Fraud	-	-	\$250,000
Telecommunications Theft	\$100,000 \$2,500	\$100,000 \$5,000	\$250,000 \$5,000
Social Engineering Fraud	\$125,000 \$12,500	\$100,000 \$10,000	\$250,000 \$5,000

ENDORSEMENTS

\$500,000 | \$5,000

\$100,000 | \$5,000

\$500,000 | \$5,000

	\$500,000   \$5,000	\$100,000   \$5,000	\$500,000   \$5,000
Preventative Shutdown Endorsement	\$500,000	-	\$50
Hardware or Equipment Betterment	-	-	\$50
Bricking Coverage	\$500,000	-	\$500,000
Bodily Injury	\$250,000	-	\$300,000

Chat

Help

© 2021 Relay Platform Inc.

Terms and Conditions

Service Level Details

Privacy Policy

# TRUFLA TECHNOLOGY

---

SENIOR SOFTWARE DEVELOPER/ARCHITECT

## REACT INSIDE of EMBER.JS

**Problem:** After the first month or so of planning and the team coming up with a grandiose vision there came the time to put theory to practice.

The first obstacle in our way was working against a fairly old version of Ember.js (2.13.1) which had zero tests without compromising the current functionality.

Of course, I was quite biased towards going down the React route, but I feel it was the only obvious practical solution without having to start from scratch which was a non-starter.

The screenshot displays an Ember.js application interface. On the left, a sidebar for a user named 'Des' (starred) shows profile details: 'Added Jul 29, 2019 · By Desmond O'Leary', Name, Company, Email (des+123@trufla.com), Phone, Website, Source (Not Provided), Address (Not provided), and Tags (No tags assigned). The main content area is titled 'React.js' and contains a 'Back to quotes' link and a form with tabs for 'Applicant', 'Cyber Risk', 'Claims & Prior Insurance', and 'Coverage'. The 'Applicant' tab is active, showing fields for 'Insured Name' (Des), 'Website' (www.google.com), 'Country' (United States of America), 'State' (Alabama), 'City' (Calgary), and 'Zip Code' (12345).

## **Major issues solved with this approach**

- Ember version was extremely out of date ○ Getting Ember.js upgraded to 3.3.x was difficult but going directly onto 3.4.x was risking the whole project
- Working within the constraints of Ember.js proved to be a development nightmare due to their overall lack of documentation and questionable design decisions.
  - Unable to upgrade fully to 3.4.x due to the number of breaking changes and risk it posed
  - Cannot compete with its competitors for community libraries and accompanying libraries
  - Ember.js are going through major changes in an effort to catch up which brings with it even more risk.
  - React is the only technology that could make this possible due to its flexibility by design to work inside any HTML

## **Major findings**

- Turned out to be a hard sell due to so many unknowns & lack of frontend knowledge within our team of three core developers.
- The biggest delays by far were compilation issues when attempting to add new libraries into the mix. Ember.js uses their own build tool namely broccoli which did not play well at all without significant effort.

## **Outcomes**

- Helped make the seemingly impossible possible in securing a massive contract.
- Updated the old look and feel to better match the new as exemplified in the image above.
- We turned zero tests into hundreds

- Released new functionality without causing any obvious issues with the old functionality.
- We continue to attract more talent to organization.

## DYNAMIC FORMS

**Problem:** Create forms as dynamically as possible in order to keep up with future needs, improve scalability and reduce maintenance cost.

In the end we went with RJSF (React Json Schema Form) on top of our own Ant design components.

### Major issues solved with this approach

- Once we built the first set of forms in order to add more you simply needed to declare a schema for each form.
- Gave us more time overall to focus on full end-to-end testing via Cypress

### Major findings

- We ended up swapping out most of RJSF components with their Ant design counterparts mainly due to the added maintenance cost of customizing their components to our needs.
- Ant Design turned out to be another unexpected coup which was made easier when RJSF introduced themes.
- The biggest pain point by far has been dealing with re-renders and conditionals but again with teamwork and effort we overcame these.

### Outcomes

- We now are able to create our forms solely based upon JSON Schemas
- We had a lot less to test so we could focus more on the full end-to-end scenarios

# HRAND/MEDI-DIRECT INC.

---

SENIOR SOFTWARE DEVELOPER/ARCHITECT

As part of architecting and developing the system I have been a ferocious learner and tried out many different technologies and techniques before coming up with a framework that is truly pleasing to the eye and is extremely easy to work with.

I work hard to ensure developers have excellent tools and helpers to ensure they can develop in a concise and consistent manner by hiding complexities under intuitive interfaces that are well documented and fully tested.

Because of this undertaken we as a team find it very easy to refactor major pieces of our codebase without having to worry about breaking anything due to how much API and maybe more importantly UI testing that hits every piece of functionality and makes things really scalable for bringing on new developers and will allows us to grow fast.

## ADVANCE BDD TESTING FRAMEWORK

**Background:** Early on in the project it was decided that we would develop with the Behavior Driven Development process. The idea was to have tests to ensure we are building the product in a way that is transparent and understandable to our stakeholders.

**Motivation:** I was finding that there was a massive amount of boilerplate and lack of maintenance when we initially started creating API and UI tests against static Behaviors. Early on I recognized this as a risk and collaborated with the team and it was agreed that I would come up with an automation process to speed up development overall and address the risk of having separate processes for the API and UI.

## [Sample Feature File](#)

I have been able to take the idea of Behavior Driven Development to the next level of automation by building a framework that does some of the following:

- All steps are implemented in a generic way in order to promote re-use across any number of resources.
- Converts background steps into fixture data which can be referenced in any other background/scenario step for convenience.
- Detailed instruction on the syntax for each step so that developers/business can come up with clean scenarios.
- A request is built up based on given input from each step which is then sent to the API endpoint and there is an automated process to verify that the data sent was actually saved correctly and audited.
- The same request is used based on changes made and is then used to control the UI and again there is a check to ensure everything is saved and audited as expected.

## [Sample Step Implementation](#)

### **Some outcomes:**

- Provides developers with a clear and concise interface that hides the complexities so they can focus a lot more the task at hand
- Extremely pluggable and adaptive to handling new resources and UI pages with minimal effort
- Provided helper methods that interact with the UI automatically based on the step e.g. if the step includes a reference it knows it will be updating an existing field.
- Simply fun to use and unbelievably well received.

# HUBSTAFF INC.

---

## SENIOR SOFTWARE ARCHITECT

My responsibilities as a software architect was to take high level requirements and come up with functional & technical requirements for each team of developers to undertake. I was working directly with the CTO and Lead architect and was able to get through an intensive trial.

Both projects (timesheets & time off features) were successfully completed within the quarter and at the same time I was working on my own side projects.

## TIME OFF

The high-level requirements here were to implement a system to allow for the creation of public holidays (paid & unpaid) and time off policies including unpaid, hourly & yearly accruals. In order to come up with a comprehensive document I did a lot of research on the problem at hand and also when through many competitors that with successful products and after multiple iterations landed on a final design. This included mockups & workflow diagrams.

## TIMESHEETS

The high-level requirements here were to create timesheets based off of activity recorded via Hubstaff client applications. Timesheets would display details of the activity including any prolonged idle time with each member with timesheets enabled expected to submit each based on pay schedule. What made this more complex was the interactions with other systems



including invoicing, billing and scheduling. With a lot of effort, collaboration and iteration I was able to come up with a detailed and approved design.

## **SSO (FRONT & BACK CHANNEL LOGOUT)**

One of the first projects I worked on as part of my trial period was to close up potential security holes with the pre-existing SSO system. This included implementing front channel logout based on the official Open ID specification. This was used when a normal logout is performed to ensure that all sessions under the same browser was invalidated.

The second piece was to implement the back-channel logout to invalidate all sessions regardless of browser by issuing a request for the identity provider to request each relying party to sign out completely.

# CISCO SYSTEMS

---

SENIOR SOFTWARE / PRODUCT DEVELOPER

*All of our major features were featured flagged to allow for continuous integration and quick rollback if a significant bug is found on production especially due to the sensitive nature of the features we were developing.*

## MODULARIZED AUTHORIZATION

One of the senior developers on extracting our login process into a separate engine and now it serves as a micro service for use across multiple applications. As part of this work we also introduced an SSO option and our own Identity Provider.

## PARTNER ACCOUNTS

Lead developer & team lead on a brand-new feature that now allows partner companies to service their clients and produce accurate reporting like never before. This is one of the highest priority projects of the year and is expected to bring in a lot of revenue

- Introduced Single Table Inheritance (STI) to differentiate major account types. Provided good encapsulation and separation of concerns.
- Devised a daily report that records summary information on all accounts sourced from MySQL, Cassandra and MongoDB
- Used the daily report to produce aggregate reports on individual partners including breakdown per customer
- Came up with a Partner Dashboard that gives them a quick look into the health of their customers

- In order to manage individual accounts, we offered the impersonation feature to allow them to manage customer accounts on behalf of the customer

## Partner Customer Dashboard

**Customers**

Search by name, GUID, email or DC MAC address

[+ New Customer](#)

Name	Connectors	Compromised	Provisioning Status	Licenses	Impersonate
Client 1	0	0%	<span>in progress</span>	0	<a href="#">Client Portal</a>
Client 2	0	0%	<span>in progress</span>	0	<a href="#">Client Portal</a>
Desmond	0	0%	<span>in progress</span>	0	<a href="#">Client Portal</a>
Desmond Client 1	0	0%	<span>completed</span>	0	<a href="#">Client Portal</a>
Desmond dasdsa	0	0%	<span>in progress</span>	0	<a href="#">Client Portal</a>
Desmond hadsdsa	0	0%	<span>in progress</span>	0	<a href="#">Client Portal</a>

No admin users to impersonate.

## Individual Partner Customer Overview

IMPERSONATING Customer #1

[Announcements](#) [Support](#) [Help](#) [My Account](#) [Log Out](#) [English](#)

Dashboard Analysis Outbreak Control Reports Management Accounts

Customer #1

Recent Events				Recent Computers			
Unknown	Unknown	2017-05-17 19:21:06 UTC		OS	Version	Hostname	Group
Unknown	Unknown	2017-05-17 19:21:03 UTC		Windows 7, SP 6.0	3.1.6.9505	Demo_CIA	Audit
Unknown	Unknown	2017-05-17 19:21:03 UTC		Windows 7, SP 6.0	3.1.6.9505	Demo_Command_Line_Arguments_Kinder	Protect
Unknown	Unknown	2017-05-17 19:21:03 UTC		Windows 7, SP 6.0	3.1.6.9505	Demo_Command_Line_Arguments_Meterpreter	Triage
Unknown	Unknown	2017-05-17 19:20:52 UTC		Windows 7, SP 6.0	3.1.6.9505	Demo_Flags	Protect
Unknown	Unknown	2017-05-17 19:20:52 UTC		Windows 7, SP 6.0	3.1.6.9505	Demo_SFicar	Audit

Recent Activities				Recent Outbreak Control Lists	
start_impersonating_user	User	desolear+customer1@csco.com	2017-05-19 21:36:13 UTC	File List	2017-05-17 19:48:36 UTC
start_impersonating_user	User	desolear+customer1@csco.com	2017-05-19 20:22:49 UTC	Exclusion Set	2017-05-17 19:48:36 UTC
start_impersonating_user	User	desolear+customer1@csco.com	2017-05-19 05:25:04 UTC		
start_impersonating_user	User	desolear+customer1@csco.com	2017-05-17 19:35:25 UTC		
update	ExclusionSet	Workstation Exclusions - Mac	2017-05-17 19:48:08 UTC		

Recent Policies	
Protect	2017-05-17 19:48:10 UTC
Audit	2017-05-17 19:48:10 UTC
Triage	2017-05-17 19:48:09 UTC
Protect	2017-05-17 19:48:09 UTC
Audit	2017-05-17 19:48:09 UTC

Applications

# GLOBAL ANNOUNCEMENTS

## Backend Details

- Implemented on top of Starburst Gem Foundation
- Includes announcement views table in order to keep track of items read by user
- On click of /announcements all announcements are viewable and marked as read

## Frontend Details

### AMP for Endpoints upgraded to version 1.1.1502678831 ×

For the list of new features and/or fixes, please see the [release notes](#) 

- Announcements (General, Maintenance [**very rare**] and Upgrades as shown above)
- Includes preview as you type

# BUSINESS CREATION REVAMP

Project to revamp how we provision user accounts

## Backend Details

Biggest issue here we've found is DB locks

- Figure out where the dead locks are coming from
- For us it was the provisioning code that was causing major issues
- Went from old attributes tables to having one table with many columns - This was done in stages
- Removed redundant items based on info on prod, many became default values that never change so no need for DB entries
- Done piece by piece until complete

- Moved provisioning to Delayed Job and ensured it worked much the same as before
- Improved the experience for the user and allowed them to do a lot more work instead of creating multiple tabs

## Frontend Details

The screenshot shows a form for creating a new user. It includes fields for Business Name, User details (First Name, Last Name, Login Email, Notification Email), Licenses (Connectors, Term), and Confirmation Emails (Send email to, Sales emails). There are also checkboxes for Incident Response and On Prem, and a Device ID field. A green 'Create' button is at the bottom right. Red error messages like 'can't be blank' are visible next to several fields.

- Unified error handling / client and server side validation looks the same whether on the form or alert

The screenshot shows a 'Sales emails' field. It contains a list of email addresses: 'desol' and 'desolear@cisco.com'. The 'desol' entry has a red dashed line underneath it, indicating a syntax error. Each entry has a close 'x' icon to its right.

Instead of just coming up with the standard fare I went a little further and still managed to deliver on time and customers loved it

- including email syntax highlighting
- validation is done using email validator that can handle multiple fields
- one click of item it goes into editing mode immediately
- x icon is highlighted on hover
- Inspired by Gmail's version

# IMPERSONATION

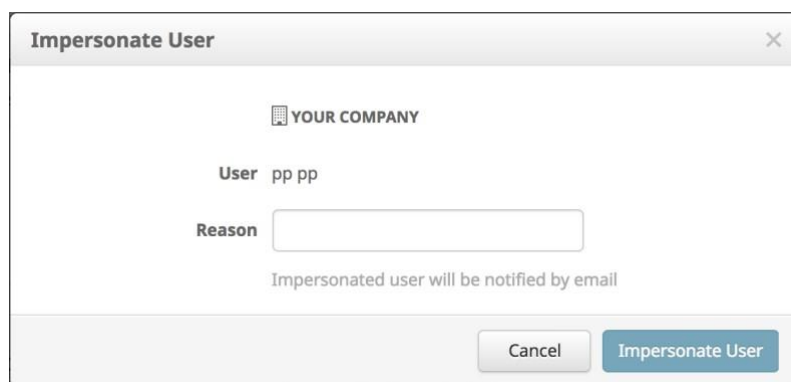
Allow our support staff to impersonate customer accounts

## Backend Details

- Found out how this is done in the wild and tested many implementations until I landed with pretender gem and implemented changes on top of this
- Audit log all actions taken by the impersonating user
- All actions by the impersonating user displayed for the customer to see
- In order for a customer to opt into this feature they had to accept an agreement

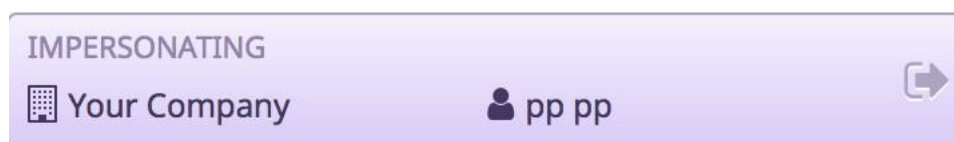
## Frontend

- Support admin only
- User has to provide reason
- Came up with widget to make it obvious we are impersonating another user



A modal dialog box titled "Impersonate User" with a close button (X) in the top right corner. The dialog contains the following elements:

- A header section with a building icon and the text "YOUR COMPANY".
- A "User" label followed by the text "pp pp".
- A "Reason" label followed by a text input field.
- A note below the input field: "Impersonated user will be notified by email".
- At the bottom, there are two buttons: "Cancel" and "Impersonate User".



## **RAILS 4 & RUBY UPGRADE**

Lead developer on Rail 4 Upgrade. Had to be done very carefully to ensure there was no customer impact.

Ruby upgrade was done with the use of feature flag and only globally enabled once all specs were passing for both Ruby versions. Rails 4 upgrade was done in a similar fashion. Ended up with no customer issues.

- Beefed up tests to ensure functionality works the same before and after the upgrade
- Replaced deprecated features
- Upgrade gems
- Fix broken specs

## **BOOTSTRAP 3 & SITE WIDE LOCALIZATION**

Lead developer on our new multi-language offering which involves localizing text across the application including handling of constants stored in the database.

Heavily involved in migrated from Bootstrap 2.x to 3.x without any production issues raised. Genuine massive undertaken that was achieved due to excellent collaboration between multiple teams.

This gave us the opportunity to clean-up the overall UI to use the newer look and feel that we introduced throughout the application. This was done with extensive collaboration with multiple team members and we came up with FAQ wiki and task list to complete the work on time and without any customer issues.

- Created a FAQ wiki in order to keep learnings / tips in the one place (this made it easier to bring others on to the team)

- Used resources that helped pinpoint necessary changes from bootstrap 2 to 3
- Feature flagged
- Formulated another FAQ for tips on adding text to localization files, this helped tremendously in keeping the workload down to a minimum for the translators