



## Case study

# Optimizing and Securing a SaaS Platform by Migrating to Angular

Some time ago, Apriorit developed a SaaS human resource management platform for our client. Since then, we have continued to collaborate on enhancing and maintaining their software. When our client needed to ensure their platform remained secure and up to date, they turned to us once again to migrate from AngularJS to Angular.

Apriorit specialists prepared the entire plan for migration from AngularJS to Angular and started implementing it, ensuring a smooth transition while optimizing and securing the platform. This allowed the client to focus on their core business functions, confident that their software was relevant, robust, and competitive.

## The client

[Payday Resources \(PDR\) Group](#) is a US-based company headquartered in Lehi, Utah, that provides a wide range of administrative services for effective employee management. In particular, PDR Group delivers online services for payroll processing, human resource support,



health benefits administration, and more.

The client owns and operates a Microsoft Azure-based HR management SaaS platform built and supported by Apriorit. Several years ago, the client [outsourced platform development to us](#), asking us to create reporting functionality, user-friendly features, and support for different subscription types.

This time, they needed help migrating their software to Angular, as support was about to end

## The challenge

The client wanted us to analyze available methods for migrating their application from AngularJS to Angular. We also needed to develop an AngularJS to Angular migration plan and start implementing it.

The client requested that we:

- Avoid rewriting the entire codebase, which is over 140,000 lines of code
- Allow bug fixes and development of new features during migration
- Ensure a gradual migration approach, keeping the platform fast, secure, and user-friendly

## The result

By leveraging our expertise, the client was able to conserve their resources and obtain a high-quality, well-structured, and ready-to-implement migration plan. We have already started implementing this plan while developing new features at the same time. As a result of our collaboration, we can now keep the client's platform secure and high-performing by ensuring that the underlying technologies remain up to date.









The existing code continues to function as it did before the update. By eliminating outdated dependencies, we managed to reduce potential vulnerabilities. The new modular system and the transition to TypeScript, CLI, and other Angular tools have significantly simplified the application's development and maintenance, expanding the possibilities for code reuse. It will also now be easier for the client's in-house team to work with the software going forward.

## Our solution

After evaluating available migration methods, we chose to use the ngUpgrade library, which enables parallel operation of both the old and new frameworks and facilitates data exchange between them. We also decided to set up a build process to support the hybrid application until migration is completed.

Since Apriorit specialists worked with this client before, we already had a connection with the in-house team. For this project, our manager formed a dedicated team consisting of a project manager, software developers, and a QA specialist.

## Project details

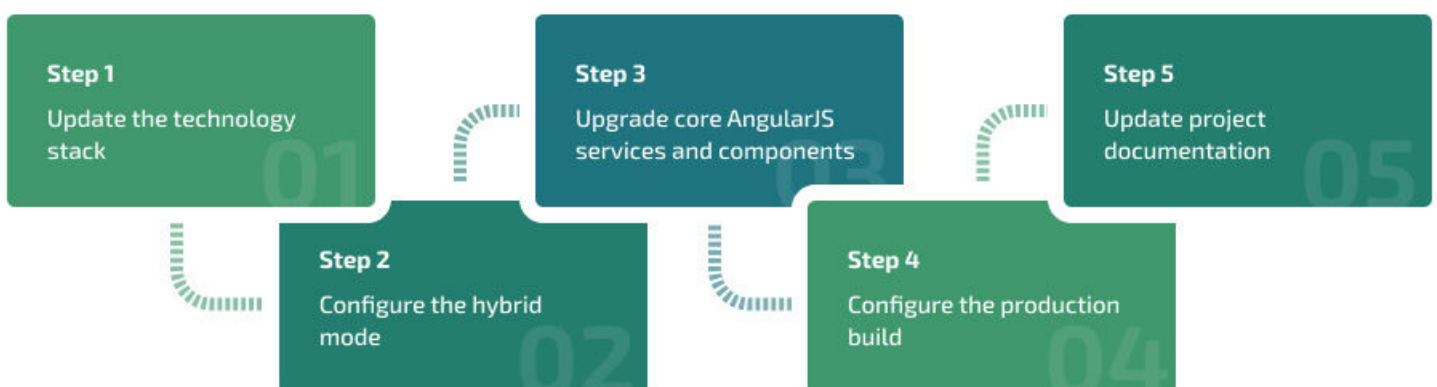
OUR TEAM	TECHNOLOGY STACK
 Frontend developers	  <ul style="list-style-type: none"><li>• JavaScript</li><li>• TypeScript</li></ul>
 QA specialist	 <ul style="list-style-type: none"><li>• Angular</li><li>• Angular CLI</li></ul>
 Business analyst	
 Tech lead	
 Project manager	

The Apriorit team communicated closely with the client's in-house team. We had regular meetings with the client several times a week and communicated via email, and discussed urgent matters on Skype and MS Teams during the migration. The internal team had daily stand-ups, planning meetings, and grooming sessions according to a Scrum approach. We managed tasks in Jira and stored documentation in Confluence.

## How we did it

We divided the project into five steps, including migration planning, preparation, and further implementation. Here's the roadmap that helped our team create a more modern and secure version of the product:

### Project stages



Now, let's take a look at our AngularJS to Angular migration step by step.

## Step 1. Update the technology stack

Apriorit experts thoroughly investigated breaking changes associated with each library being updated, including AngularJS, UI-Router, and third-party libraries. We defined strategies to address these changes, developed a comprehensive update plan, and executed it successfully.

By doing so, we were able to ensure compatibility with the libraries required for migration. Our developers also eliminated outdated code and dependencies and addressed legacy bugs that were present in the existing product version, improving the application's performance and security.

## Step 2. Update the technology stack

During this step, the Apriorit team successfully set up a new Angular application as the basis for migrating the existing AngularJS app. Here is how we did it:

- **New Angular application.** We created a fresh Angular application using the Angular Command Line Interface (CLI).
- **Code migration.** Our developers moved the AngularJS code to the new Angular project structure.
- **Package installation.** We added the packages required for the migration process.
- **Hybrid mode.** We set the application to support both Angular and AngularJS simultaneously during migration.
- **AngularJS compilation.** Apriorit engineers configured the Angular CLI to build the AngularJS code.
- **Configuration.** We managed global styles and libraries using angular.json.
- **Development server.** Our developers set up Secure Sockets Layer (SSL) encryption and proxying to a backend server for the development environment.
- **TypeScript for AngularJS.** We also enabled TypeScript for AngularJS components, significantly improving the productivity of working with AngularJS controllers and components that are not yet ready to be rewritten in Angular.

This new setup allows for smoother migration and provides additional benefits, such as the ability to use TypeScript for AngularJS components.

## Step 3. Upgrade core AngularJS services and components

During this step, our team successfully laid the groundwork for migrating to the new Angular framework by identifying and migrating key components and services. This process involved:

- Identifying core services needed across the application
- Determining common UI components used throughout the application
- Migrating straightforward components to Angular and integrating them into AngularJS
- Upgrading more complex AngularJS components to TypeScript and integrating them into Angular
- Preparing the application for adding new functionality using the new framework
- Enabling reuse of services and components in both Angular and AngularJS

As a result, we allowed for a gradual transition and reuse of existing code in the future.

## Step 4. Configure the production build

The team successfully configured the build process for production, which allowed us to efficiently deploy and test the hybrid application. Here is how we did it:

- Configured the system to copy static assets (such as images and CSS) to the build directory
- Automated the process of adding dependency injection (DI) annotations in AngularJS
- Modified the continuous integration (CI) process to use Angular CLI for building the application
- Deployed the hybrid application to a testing environment for evaluation

## Step 5. Update project documentation

Apriorit developers updated the documentation on how to set up a local development environment, as well as described the new project structure and the changes that we made. Essentially, we created a guide on how to migrate AngularJS to Angular, — specifically, how to

## Step 5. Update project documentation

Apriorit developers updated the documentation on how to set up a local development environment, as well as described the new project structure and the changes that we made. Essentially, we created a guide on how to migrate AngularJS to Angular, — specifically, how to migrate the existing code to Angular for the in-house team to use in the future.

By sharing knowledge within the team, we've reduced the [bus factor](#) and improved team efficiency. Now, developers won't waste time on additional communication and changes after code reviews, and it will be easier for new team members to get up to speed.

By following these five steps, we can use Angular for new features while gradually refactoring existing code, progressively reducing the reliance on AngularJS until the entire application is upgraded.

## The impact

The client has received a product update that allows them to replace an outdated framework with a modern one. Currently, the project uses a hybrid version of Angular. We are writing new product components in Angular and constantly moving old components to Angular as the opportunity arises. As a result of our collaboration, the client now can:

- Migrate their software gradually, at a pace that suits their needs
- Be sure that existing code will continue to function the same as before migration
- Replace old libraries with their Angular equivalents and use ngUpdate to integrate them into AngularJS code. This allows for the introduction of new features and functionalities to the existing application without a complete rewrite.

With the help of Apriorit, our client received the tools needed to modernize their application without disrupting their current operations. The phased approach to AngularJS to Angular migration also reduces the risk of introducing new bugs or errors during the migration process.

*Want to improve your application's security score? Contact Apriorit so we can start developing an effective security solution together!*