



Case Study

Middleware for logistics services that facilitates seamless data exchange between multiple transportation management systems

Case Study

Developed a standalone middleware solution for a leading logistics company to facilitate seamless data exchange between their transportation and data warehouse management systems.



Summary

We developed a centralized software interface to interconnect our client's VMT, TMFF, WMS, and WinWeb systems for effective transportation management. The solution was focused on facilitating real-time data exchange between the aforementioned entities via middleware. The middleware would keep on accessing all data changes from a specific directory. It will process the data as per the requirements (in terms of data structure and format), and send it to the respective system. Accordingly, it would update the status pertaining to the data exchange between the given systems at any point of time. To develop the full-scale middleware solution, we used a tech stack consisting of Spring Boot and Apache Camel. We shall have a detailed discussion on its workflow in the following section.

Industry

Logistics and Transportation

Users/customers

Large-scale retailers, electronics and automotive companies, etc.

Tech Stack

Spring Boot, Apache Camel, jBPM, PostgreSQL, Docker

Team Size

7 Oodlites
(1 Technical Architect, 1 Team Lead, 3 Backend Developers, 1 Frontend Developer, 1 QA Engineer)

Scope of Work (SoW)

Our client was a leading logistics service provider having their warehouses across multiple locations in Europe and China. They offered logistics and transportation services to customers including large-scale retailers, manufacturers, and eCommerce businesses by carrying out their shipments via air, sea, and rail freights. The client sought our services to develop a standalone middleware solution to interconnect their heterogeneous software systems i.e WMS, VMT, TMFF, and WinWeb. The SoW required us to develop the architecture for an interconnected and horizontally scalable solution that would facilitate uninterrupted data exchange between the given systems. Below are the features and functionalities that were requested by our client.

1. Middleware Solution

We were asked to develop a robust integration engine to facilitate frictionless data exchange among their VMT, TMFF, WinWeb, and WMS systems.

2. A Robust Microservice Architecture For Data Handling

Data exchange between heterogeneous systems was the core requirement of the client for the given project

3. Extensible APIs For Easy Pluggability

We were supposed to develop extensible APIs that could be integrated with other modules as and when required

Solution

We analyzed our client's requirements and had a kick-off meeting to put together a team of developers for the project. Our team began with a careful assessment of the client's existing business model and formulated an execution strategy to achieve the tasks at hand. We followed agile methodologies to carry out the development process in a phased manner, starting with graphic design and then moving ahead with the frontend and backend interfaces. We also deployed a QA engineer on time-and-material basis to test each functionality against a given set of quality standards.

We successfully achieved our client's requirements with the following deliverables:

1. Built The Middleware Using Apache Camel

Apache Camel is an open-source framework for message-oriented middleware with a rule-based routing and mediation engine. We used this framework to develop a comprehensive integration engine that facilitates seamless communication between heterogeneous systems for effective transportation management.

2. Developed a Microservice Architecture Using Spring Boot

We used the Spring Boot framework to develop a microservice architecture to render increased flexibility and greater resilience to the code. It would further provide better maintenance, improved performance, and greater fault tolerance for effective business alignment.

3. Established a Robust Database Management Architecture

To address our client's requirements, we implemented PostgreSQL as the primary and persistent database for effective storage and manipulation of data. The main reason for using PostgreSQL was to provide all SQL features like indexes, constraints, triggers, and views in addition to some NoSQL features as well.

4. Developed The Web and Mobile Applications



We provided complete support for web and mobile application development for the middleware. Our team worked consistently to develop the entire frontend and backend of the application with multiple sprints. After developing their web application, we used native mobile frameworks to build their Android and iOS apps.

5. Configured The Application For User Authentication

Our team configured the application to facilitate a secure user authentication process for both VMT and TMFF systems. It included configuration of the user login credentials, storing authToken for different users, and sharing authToken in each API call.

6. Exporting and Revoking Milestones

We developed dedicated modules for exporting and revoking information from TMFF and feeding it to the middleware. At the same time, it would update the status in the VMT system after converting the information into the VMT dashboard format.

7. Application Deployment Using Docker

We used Docker for deploying the application over the cloud in a series of production-ready containers. It significantly reduced the deployment time and rendered a lightweight software environment as compared to virtual machines.

Results

We successfully completed the project within the stipulated time frame and delivered the following results:

1. Developed the middleware solution (using Apache Camel) to virtually interconnect our client's transportation management systems
2. Provided assistance for the domain and architecture design of the middleware
3. Developed the web and mobile application along with the complete frontend and backend interfaces
4. Established a complete database management architecture for storage, retrieval, and manipulation of data at any given time
5. Provided extensible APIs for easy integration with the required modules
6. Developed a microservice architecture for easy application deployment over the cloud
7. Deployed the application over the cloud in a series of containers using Docker

About Oodles ERP

Oodles ERP is a software services company that offers complete enterprise software development services with a focus on implementing next-gen technologies. With a proven track record in custom ERP development, we have successfully completed 50+ software projects related to CRM, HRM, inventory/warehouse, eCommerce, supply chain, and logistics. We are mainly focused on helping startups and small-to-medium enterprises to achieve digital transformation through cost-effective ERP software solutions.

Follow Us



erp@oodles.io