

 Success story

# Enhancing performance and collaboration for a global freight forwarder

Our client, a leading global freight forwarding company, encountered significant challenges in scaling their project as it outgrew their capacity to manage it effectively. VirtusLab implemented a monorepo structure, optimized the build process by migrating it to Bazel, and enhanced testing accuracy. This transformation led to faster build times, improved system performance, and a more streamlined development process.

 **The Challenge**

Our client aimed to manage a complex logistics data project that had outgrown its original scope. The project, which had started with a limited feature set, expanded over three years to handle millions of messages daily, requiring our client to resolve several issues.

- **Scalability:** Our client needed a system that could support new features and a growing codebase without performance degradation.
- **Build and Compilation Times:** As project components grew, the company's build tool, sbt, took up to 60 minutes to compile. Also, the tests were slowing down CI/CD.
- **System Performance:** To process the data without bottlenecks, the system required high performance. Initially, the use of runtime serialization had slowed the system and caused test timeouts.



- **Collaboration and Code Management:** As the codebase expanded, managing the project across multiple teams became challenging. Choosing between monorepo and multi-repo solutions was critical to maintaining productivity and quality.

Our client lacked the manpower to maintain the progress of development and meet scaling needs. They turned to VirtusLab, due to our experience with monorepos and improving build times.



## The solution

VirtusLab implemented several solutions to meet the freight forwarder's technical and operational requirements.

- **Monorepo structure:** Our engineers introduced a monorepo approach to streamline code sharing. It also allowed them to reduce the unnecessary workload that would normally result from managing multiple repositories. This approach centralized the code, enabling easier access and collaboration across teams.
- **Optimized code generation with Scala macros:** To address deserialization issues, VirtusLab moved code generation to the compilation stage with Scala's macro capabilities. Doing so reduced runtime load and helped catch errors earlier.
- **Enhanced developer productivity:** VirtusLab improved developer workflows by introducing code-generation support, multi-stage builds, and caching, reducing build times thus enabling faster code-run loops.

## Migration to Bazel

VirtusLab migrated the build system from sbt to Bazel, which offered several key advantages.

- **Parallelized builds:** Bazel's parallel processing reduced build times significantly.
- **Remote caching:** Cached builds improved speed by reusing previously compiled code.
- **Selective compilation and testing:** Bazel's selective processes enabled faster, more focused compilation and testing.
- **Unified tool for frontend and backend:** Bazel supported the project's frontend, making it a single-build solution across the system.

Our engineers maintained a parallel build setup with Bazel and sbt during the migration. This allowed for artifact verification reduced risk, and preserved development progress while Bazel was integrated.



## ★ The results

VirtusLab's solutions yielded improvements across the project's performance, productivity, and scalability.

- 1 **Reduced build times:** Build times decreased from 40-60 minutes to as low as 5 minutes, Some CI/CD builds were completed in 1 minute.
- 2 **Improved system performance and reliability:** Moving code generation to the compile stage has freed the process from its bottlenecks, improving system performance and reliability. This change allowed the system to process and deserialize messages efficiently, even under heavy workloads.
- 3 **Simplified code management and collaboration:** The monorepo structure streamlined code sharing and dependency management across teams, reducing friction.
- 4 **Accurate testing with fewer false positives:** Bazel's testing capabilities minimized flaky tests, improving accuracy and reducing unnecessary test alerts.
- 5 **Optimized developer experience:** Bazel's fast local builds allowed developers to work without relying on persistent sbt servers, resulting in higher productivity and satisfaction.

## The tech stack

/ Programming language:



/ Frameworks and libraries:



/ Build and development tool:



/ Other tools:



# About VirtusLab

At VirtusLab, we aim to lead in software technology, working consistently to enhance efficiency. Our profound commitment to research and development and a dedicated focus on emerging trends and inspirations fuels an innovative culture. This ethos precisely guides advancing our cutting-edge solutions, inviting collaboration to expand the boundaries of software technology collectively. We welcome you to be a part of this transformative journey.

[Let's connect](#)

## Contact Details

[info@virtuslab.com](mailto:info@virtuslab.com)

**/ POLAND**

**Kraków Headquarters**

Virtus Lab Sp. z o.o.  
ul. Szlak 49  
31-153 Kraków

**/ GERMANY**

**Berlin Office**

**+49 30 52014256**  
VirtusLab GmbH  
Potsdamer Platz 10  
10785 Berlin

**/ UNITED KINGDOM**

**London Office**

**+44 (0)20 4577 1051**  
Virtuslab Ltd.  
40 Bank Street HQ3  
London E14 5NR