

# Legacy Modernization For a Software Licensing Company

How a security tech manufacturer cut costs on the system maintenance and gained ISO 27001:2013 certification

Industry: Technology

JAVA DEVELOPMENT

LEGACY SOFTWARE MODERNIZATION

## Business Challenge

### What issues were restraining our client?

When it comes to modernizing legacy systems, companies tend to put it off until the very last minute. The reasons for that might be different: additional expenses, difficult orchestration of the initiative, resistance to change... However, the risks arisen from doing nothing are equally high for everyone and there usually comes the point when hanging on to the old software is no longer possible.

That's what happened to our client, a global licensing software developer. They delayed updating their back-end software until they had an urgent need for ISO27001:2013 certification. And it hit in 2022 when the updated certificate version, ISO27001:2022, came out.

That was where our client needed expert support. They created their licensing software in the early 2000s. Those were unique, state-of-the-art solutions, which worked securely even 15 years after the release. When the client held a hacking contest to test and evaluate the system's reliability, none of the participants succeeded.

However, with time, some technologies tend to be replaced by others, and legacy software is no longer supported by the provider and doesn't receive security updates, becoming your Achilles heel. The same situation became the main driver for legacy system modernization at our client's organization.

The client product was based on the **outdated SOAP API and legacy Apache products**, such as the Axis2 framework and the TomEE 7 application server. TomEE 7 stopped receiving security updates in 2021, so the client updated it to the next version to get ISO 27001 certification. However, this change led to other ones, as TomEE 8 wasn't compatible with the Axis2 framework they used.

At this point, the client realized that dealing with those changes would be beyond their capacity and capabilities. Hence, they started looking for a company with expertise in legacy system modernization.

The task was complicated as they had never outsourced software development and finding a software company that would fully meet their requirements was problematic.

### What were the client's mandatory requirements for the future tech partner?

**01 Location in Europe and compliance with European security regulations.** The client was looking for a technical partner among European companies that follow strict data security regulations and hold ISO 27001:2022 and ISO 9001:2015 certifications. It was also crucial to be in the same time zone to quickly resolve working issues.

**02 Expertise in both legacy and up-to-date software.** Finding a developer who can work with Axis2, as this framework is obsolete and rarely used, was a tough legacy system modernization challenge.

**03 High security demands to the development team.** The client was looking for a team that would work with their software on-site by using keys (dongles) that would be sent to a flash drive.

Having a wide array of strict requirements, the client chose to entrust their fulfillment to \*instinctools as a global software development company with extensive expertise in delivering innovative solutions since 2000. Over this time, we've built a skillset based on a diversity of technologies ranging from PHP to Python and React. Given the talent crunch in the modern IT industry, such an extensive expertise is truly appealing to our clients.

Moreover, our headquarters is located in Germany, and we follow strict European regulatory requirements for data security, such as GDPR, EDPB, DPLE directive, etc.

Our team can also work as an [offshore development center](#) providing a secure physical perimeter for the project.

Thanks to competitive advantages of our company, we passed the client's rigorous selection process.

To enable the client to get ISO 27001 certification, we needed to modernize legacy systems, for instance, the SOAP API through which the requests such as license creation were sent. For this purpose, we decided to upgrade the existing back-end system under the SOAP API to receive regular security updates.

## Solution

### Moving fast but radically or slowly but surely?

After evaluating the project, we offered the client two legacy system modernization strategies to choose from.

#### Renovating the existing software to the most up-to-date solution.

The most advanced option would be Spring. It's an open-source Java-based framework that empowers developers to build high-performing applications. It is currently the standard of [enterprise software development](#), with no analogs, as Spring can cover all the needs of enterprise out of the box. Moreover, it's an actively developing framework, which implies the constant contribution of new features and security checks.

#### Going step by step by upgrading the versions of existing software and, eventually, coming to Spring.

If a client isn't ready for substantial infrastructure changes (which, for instance, Spring requires), they can move gradually. With this approach, they could pass ISO 27001:2013 certification, and while it's valid, deal with the rest of the innovations and implement Spring at a comfortable pace. And with the updated technology stack, they'll be able to go through ISO 27001:2022 certification later.

Since this was the client's first experience with an outsourcing company, they wanted to move in small iterations and decided to follow an incremental legacy system modernization approach.

It's also important to note that our client has been creating IT solutions for software licensing for over 30 years, and **their development team has stayed the same** from the moment of the product release. On the one hand, it was helpful because the client's team knew the product down to the last detail. On the other hand, they got used to relying on outdated technology stacks and approaches. To switch straight to a cutting-edge solution, the client would have to invest a lot of time in knowledge transfer to their specialists.

Therefore, the client chose a more granular change so that they could reap the benefits of modernizing legacy systems, with the team gradually adapting to the new environment and gaining knowledge about working with state-of-the-art technologies.

Before turning to \*instinctools, the client had already upgraded TomEE 7 to the 8th version. However, it was later discovered that TomEE 8 couldn't work with Axis2 and requires another framework. Since Axis2 is an Apache product, we offered to switch to another Apache framework – CXF that has replaced Axis2 as a more present-date solution.

Nevertheless, code migration from Axis2 to CXF without changes is impossible as these frameworks have dissimilar approaches to code writing. Axis2 works with abstractions and classes, while CXF works with data models. Therefore, besides code migration, we tailored the code to the new framework.

We did code refactoring and uncovered parts of the code for which more elegant solutions could be used.

### Going beyond expectations

If we see an opportunity to do more for the client, we never hesitate to go for it. In the case of legacy system modernization for our client, we provided them with a few extra-mile solutions.

#### ■ Updating Java

The client had Java 8, which was compatible with TomEE 8 and CXF. However, Java 11 contains an updated encryption protocol (TLS 1.3 instead of TLS 1.2), which is faster and ensures a higher security level, so we installed Java 11.

#### ■ Implementing annotations for authorization service

The legacy client authorization model was based on roles, where each user had their own role with specific access rights. We switched it to a new model based on annotations. This legacy system modernization approach significantly reduces the amount of code and simplifies maintenance, as clear code written according to modern standards is easier to maintain.

Moreover, familiarity with annotations is another small step toward Spring implementation in the future. This framework extensively uses annotations, and if you haven't encountered them before, it's hard to figure them out.

#### ■ Detecting bugs in the client's testing system.

For the client, one of the criteria that the legacy system modernization project was done correctly was to pass their tests. They provided us with their test infrastructure. However, we found bugs in those tests, such as PHP clients connected to an outdated framework. We reported them to the client so that they could fix them.

Along with upgrading legacy systems, we trained employees on the client side to ensure knowledge transfer so that they could feel confident working with CXF and TomEE 8.

We've carried out legacy system modernization for the client considering upcoming ISO27001 certification and the switch to Spring in the future.

The client was successfully certified and received ISO 27001:2013 in November 2022.

By that point, our project had been over, but we consulted the client on the next steps of their legacy software modernization. As they were totally satisfied with our collaboration, they decided to keep working with \*instinctools in the offshore development center format and gradually move toward Spring to adopt it by 2025. By that time, the ISO 27001:2013 certification will have expired, and with the updated software, they will be able to obtain ISO 27001:2022 certification. So at the moment, we have two more legacy system modernization projects with the client and look forward to bringing them the same value as in the first project.

## Key features

#### Flexibility

Refactoring the code according to modern standards allows the client to expand the range of specialists they can hire as more developers can work with clean code.

#### Scalability

Switching to an up-to-date framework and application server has made our client more adaptable and prepared for subsequent steps of their legacy system modernization journey.

#### Modernization

Transition to an authorization model based on annotations has made the upkeep of the system easier

## Business Value

- Software modernization for **ISO 27001:2013 certification** obtainment
- The backbone for future **Spring adoption**
- Simplification and **cost reduction** for staffing and maintenance

## Multiplier Effect

Legacy system modernization is valid for any industry. However, the SOAP API, which is the basis of the legacy system in the case of our client, is prevalent in [banking](#). For banks that have existed for more than a couple of decades, an abrupt transition to up-to-date software is also challenging because of drastic infrastructural changes and time for knowledge transfer.

And that's ok! Switching to a cutting-edge REST protocol just because now it's considered a standard is not the only option available. We don't swing to the buzzwords and choose the most appropriate solution considering your requirements and limitations. As in the case of our client, you can stay on SOAP but upgrade your backend software to meet your actual needs and ambitions.

### Do you have a similar project idea?

Contact us — and we will estimate your projects costs for free!

CONTACT US