



REMAINING COMPETITIVE WITH APPLICATION MODERNISATION

**BUILDING SMART SOLUTIONS
THAT CREATE BUSINESS IMPACT**

Rebuilding existing applications with
more modern technologies

REMAINING COMPETITIVE

Rebuilding existing applications with more modern technologies is important to remain competitive. Many large enterprises still rely on legacy applications to run their business operations. These legacy systems are built on programming languages that are outdated and hard to maintain which creates many challenges. It is difficult to find engineers who know how to provide support for old technologies, the cost for maintaining can be too high, the UI can become non-responsive, or the customers are asking for more.

When it comes to modernising a legacy application for your business, there are a few important questions that you need to consider.

- What are the costs of maintaining your legacy application?
- What resources do you need to support it?
- How can you find the people who can support it?
- How would you rate the performance of your legacy application?
- Are your customers expecting more from your application?
- Is your legacy system able to handle the growth of your business?

Legacy systems can be extremely large and difficult to maintain. The hardware that supports the application can also become obsolete and incompatible with modern IT systems while the engineers who know how to support it retire.

Most of our clients are enterprises in sectors like finance, banking, e-mobility, and logistics. Enterprises in heavily-regulated industries find it challenging to modernise their applications, which is why having a trusted partner who has the experience and the knowledge is essential. Examples of applications that we at Strypes modernise include tailor-made CRM, time and resource management systems that require us to develop custom-made solutions.

As a tough partner, we at Strypes offer application modernisation as a continuous service. We help our clients assess the work that needs to be done and then develop a solution to modernise their application. Typically, this results in increased customer satisfaction, smoother operations, increased performance, and reduced costs.

THE DIFFERENT LAYERS OF APPLICATION MODERNISATION

UI/UX Modernisation

Improving the overall look and feel of the application can help enterprises make their operations smoother, meet the requirements of their customers, or attract new clients. As UI/UX modernisation is focused on the front-end, it won't alter the behaviour of a legacy application. To do that, we can look at the migration solutions mentioned below.

Code Migration

This type of technology upgrade migrates the code to a more modern language (Java, C#, etc.) or to a recent version of the same stack (.Net Core, Entity Framework, SQL, etc.). Code migration replaces the application-specific features with their equivalents in the new environment. However, this type of partial modernisation doesn't change the monolithic architecture which keeps companies unable to meet today's data demands.

Database Migration

With this approach, data is migrated to a more modern database management system which makes the application more robust. The client will notice how their application has become much faster and that it works better, without changing the overall look and feel. Database migration won't modernise the UI, architecture or technology of the legacy application.

Cloud or Server Migration

Companies can shift from hosting their application on their on-premise servers to the cloud which can improve the uptime and ensure better data security. Cloud migration, however, may not be suitable for companies that are under strict regulations, even when the level of security is the same, if not higher than their on-premise servers. Server migration is another solution that can increase performance and security, although it requires companies to have their own cybersecurity teams.

The Full Package

This type of technology upgrade migrates the code to a more modern language (Java, C#, etc.) or to a recent version of the same stack (.Net Core, Entity Framework, SQL, etc.). Code migration replaces the application-specific features with their equivalents in the new environment. However, this type of partial modernisation doesn't change the monolithic architecture which keeps companies unable to meet today's data demands.

CASES WHERE APPLICATION MODERNISATION IS NEEDED

Usually, our clients come to us when they have one or several of the following issues:

The UI becomes non-responsive

The UI/UX is not intuitive or aligned with the expectations of the modern customer

Changes in the system are costly and require risky redeployment

Lack of engineers who know how to support the legacy technologies

Inability to scale or innovate due to code complexity, incompatibility, or lack of flexibility

The application becomes unstable

Software modernisation is a continuous service and in some of these cases, outsourcing it makes good business sense. As a thinking partner, Strypes can provide the right team, build the solution, and maintain the knowledge of your legacy system.



INTUITIVE UX/UI



STABILITY



PERFORMANCE

CASE STUDY: RESOURCE MANAGEMENT SYSTEM MODERNISATION

Strypes recently created a prototype to modernise the resource management system of a big port (under NDA). Although the port was not our client, we developed a solution as part of our project proposal to them. Our goal was to improve the overall experience and reduce the maintenance cost, transitioning from a desktop to a web application. For this project, we used a layered approach to modernise the front-end, the code, and the database.

The port had two major challenges:

At some point in 2019, screens (more than 90) of their resource management application had started to black out as the system had become unstable.

It was difficult to find engineers who could support the legacy technologies and software. This was because these technologies were created in the 80s and most engineers who knew how to support them had retired.

The application was programmed in C++, in Dutch, and its role was to manage the personnel of the whole port – schedules, educational activities, skills, vacation and sick leave, and general resource management. One of the requirements for this project was to keep the same look and feel of the old application. There was a large number of people who worked for the port and they were used to the interface of the legacy resource management system.

THE TECHNOLOGICAL STACK WE USED

Using our expertise, we developed the prototype using .Net Core and Entity Framework, migrating the code from C++ to C# and the database from Oracle to SQL Server. To modernise the UI/UX, we used the Aurelia framework and JavaScript.

While the old application needed to be compiled, installed, and configured. To load the modernised application, the staff had to just type the URL in the browser.

LEGACY TECHNOLOGY

C++

Oracle RMB

VAX Machines

NEW TECHNOLOGY

.Net Core, C#

SQL Server

Aurelia, JavaScript

As a thinking partner, we always give recommendations to our clients and in this case, we also recommended cloud migration. However, as the port was working with information that was very sensitive, they had to comply with various regulations and certificates. This is why they made the decision to host all the information on their on-premise servers.



NEXT STEPS

Application modernisation is usually a long-term partnership, especially for fast-growing companies that are more dynamic and innovative. Being a thinking partner puts us in a position where it is essential to understand the processes and the problems of our clients. We keep the knowledge and we usually remain responsible for supporting the application after we have modernised it.

REQUEST A CONSULTATION



The examples that we reviewed showcase a small fraction of how to apply Nearsurance as an outsourcing model that guarantees a solution.

The best way to show what we can do for your company is to offer a free consultation so that we can discuss your needs and show you what solutions we can build.

To request a demo, reach out to us at business@strypes.eu or call us to discuss your project needs.

Visit our website

Visit <https://strypes.eu> to read more about our services.



Miroslav Grozev
Group Manager

Has been with Strypes since 2009



George Stoyanov
Software Developer

Has been with Strypes since 2019

NETHERLANDS

Kopenhagen 9
2993 LL Barendrecht
E: business@strypes.eu
M: +31 6 156 883 17

NETHERLANDS

Prof. Dr. Dorgelolaan 30
5613 AM Eindhoven (4th fl)
E: business@strypes.eu
M: +31 6 814 081 88

GERMANY

Fraunhoferstr. 9
85737 Ismaning
E: business@strypes.eu
M: +49 89 995 29570

BULGARIA

10 A, Maystor Aleksii Rilets
Sofia 1618, Bulgaria
E: business@strypes.eu
M: +359 893516833