ESETTANULMÁNY · FRONTX KFT.

Legacy Java back-office alkalmazások konténerizációja

Hogyan tettük hordozhatóvá és egyszerűen telepíthetővé egy állami nagyvállalat két legacy Java nyilvántartó alkalmazását - az üzleti logika érintése nélkül.

Ügyfél	Magyar állami nagyvállalat
Projekt típusa	Legacy modernizáció, konténerizáció
Időtartam	~5 hónap, teszteléssel együtt
Csapat	FrontX: projektmenedzser, adatbázis-architekt, Java fejlesztő
Fő technológiák	Java 11, Payara Server, PostgreSQL, Docker, Git, Nexus, Linux

1. Az ügyfél és a kontextus

Megbízónk egy magyar állami nagyvállalat, amelynek napi működését két legacy Java alapú back-office nyilvántartó alkalmazás támogatja. Az alkalmazások Java 11, Payara alkalmazásszerver és PostgreSQL adatbázis környezetben futottak, Windows szervereken.

A cél az volt, hogy az alkalmazásfuttató környezet Windowsról Linuxra kerüljön - úgy, hogy a bevált üzleti működés ne változzon, az üzemeltetés pedig egyszerűbbé váljon.

2. A kihívás

A FrontX tervezőként és kivitelezőként vett részt a projektben. A kiinduló helyzet több szempontból is nehéz volt:

- **Kötelező platformváltás:** az alkalmazásfuttató környezetet Windowsról Linuxra kellett migrálni
- **Költséges natív migráció:** a két alkalmazás natív átültetése az új környezetre nagyon jelentős munkát igényelt volna
- **Lassú működés, nehéz üzemeltetés:** az alkalmazások nagyon lassan futottak, és nehezen voltak üzemeltethetők
- **Beégetett környezeti beállítások:** az SP integráció, a mobilalkalmazás-integráció, a fájlszerver elérési útvonalai és a szerverkörnyezeti beállítások natívan, a kódban voltak rögzítve, így minden környezeti változás fejlesztői beavatkozást igényelt

3. FrontX megközelítése

A FrontX részéről egy projektmenedzser, egy adatbázis-architekt és egy Java fejlesztő dolgozott a projekten. A fejlesztés során AI-alapú fejlesztői eszközöket is használtunk. A megvalósítás a teszteléssel együtt körülbelül öt hónapig tartott.

Konténerizáció natív migráció helyett

Az alkalmazások natív átírása helyett Docker-konténerbe helyeztük őket. A Payara alkalmazásszerver, a PostgreSQL és a további futtatókörnyezeti elemek a konténeren belül futnak, így a megoldás hordozható, és a hosztoló operációs rendszertől függetlenül működik.

Egy közös Docker réteg két alkalmazáshoz

A két alkalmazáshoz ugyanaz a Docker réteg készült el, oly módon, hogy az üzleti logikák változatlanok maradtak. Ez csökkentette a fejlesztési és a későbbi üzemeltetési ráfordítást, és egységessé tette a két rendszer kezelését.

Git-Nexus integráció, next-next telepítés

A konténerizált megoldás Git és Nexus integrációval épül fel, így a telepítés egyszerű, lépésről lépésre végigvihető (next-next) folyamattá vált.

Környezeti beállítások kiszervezése

A korábban beégetett környezeti beállítások - SP integráció, mobilalkalmazás-integráció, fájlszerver elérési útvonalak, szerverkörnyezeti beállítások - egy külső setup fájlba kerültek, ahonnan egyszerűen módosíthatók.

4. Eredmények

~5 hó	2	1	0
teszteléssel együtt	legacy Java alkalmazás	közös Docker réteg	üzleti logika módosítás

A két alkalmazás natív migráció nélkül került át az új környezetbe. A projekt eredményeként:

- **Gyors telepíthetőség:** a Git-Nexus integrációnak köszönhetően a telepítés gyors, next-next jellegű folyamattá vált
- **Környezetfüggetlenség:** az alkalmazások a teljes futtatókörnyezettel együtt konténerben futnak, így hordozhatók, és nem függnek a hosztoló operációs rendszertől
- **Jobb teljesítmény:** a korábban nagyon lassú alkalmazások a konténerizált környezetben jobb teljesítménnyel működnek
- **Változatlan üzleti logika:** a bevált üzleti működéshez nem kellett hozzányúlni, ami csökkentette a projekt kockázatát
- **Egységes megoldás:** mindkét alkalmazás ugyanarra a Docker rétegre épül
- **Konfigurálható környezet:** a környezeti beállítások a külső setup fájlból, fejlesztői beavatkozás nélkül módosíthatók

5. Tanulságok - amit más cégeknek is javasolunk

A projekt tapasztalatai különösen azoknak a szervezeteknek lehetnek hasznosak, amelyek régi, de üzletileg fontos alkalmazásokat kell új környezetbe vinniük:

- **A platformváltás nem feltétlenül jelent újraírást:** konténerizációval egy legacy alkalmazás új környezetbe vihető anélkül, hogy a natív migráció teljes költségét és kockázatát vállalni kellene
- **Egy közös réteg több alkalmazást is kiszolgálhat:** ha a futtatókörnyezetek hasonlók, egyetlen jól megtervezett Docker réteg több rendszer modernizációját is lefedheti
- **A konfigurációt érdemes kiszervezni:** a beégetett környezeti beállítások külső fájlba helyezése a modernizáció mellékhatásaként is komoly üzemeltetési előnyt hoz
- **Az AI-eszközök a legacy munkában is segítenek:** a fejlesztés során használt AI-alapú eszközök a legacy környezetben is támogatták a csapat munkáját

A FrontX-ről

A FrontX Kft. professzionális IT projektmenedzsment és egyedi szoftverfejlesztés szolgáltatója. ~20 fős csapatunk 20 éves nagyvállalati tapasztalatra épít – pénzügyi szektor, biztosítók, enterprise IT projektek területén. Segítünk megvalósítani digitális elképzeléseidet, a tervezéstől az éles üzembe helyezésig.

frontx.hu · info@frontx.hu · Behán László, ügyvezető: +36 20 938 2448

CASE STUDY · FRONTX KFT.

Containerising legacy Java back-office applications

How we made two legacy Java registry applications of a state-owned enterprise portable and easy to deploy - without touching the business logic.

Client	Hungarian state-owned enterprise
Project type	Legacy modernisation, containerisation
Duration	~5 months, including testing
Team	FrontX: project manager, database architect, Java developer
Key technologies	Java 11, Payara Server, PostgreSQL, Docker, Git, Nexus, Linux

1. Client & context

Our client is a Hungarian state-owned enterprise whose day-to-day operations are supported by two legacy Java back-office registry applications. The applications ran on Java 11, the Payara application server and a PostgreSQL database, on Windows servers.

The goal was to move the application runtime from Windows to Linux - without changing the proven business operation, while making operations simpler.

2. The challenge

FrontX took part in the project as both designer and implementer. The starting point was difficult in several respects:

- **A mandatory platform change:** the application runtime had to be migrated from Windows to Linux
- **A costly native migration:** porting the two applications natively to the new environment would have required a very significant effort
- **Slow performance, hard to operate:** the applications were very slow and hard to operate
- **Hardcoded environment settings:** SP integration, mobile app integration, file server paths and server environment settings were hardcoded, so every environment change required a developer

3. FrontX's approach

On the FrontX side the team was a project manager, a database architect and a Java developer. AI-based development tools were also used during the work. Delivery, including testing, took about five months.

Containerisation instead of native migration

Instead of rewriting the applications natively, we placed them in Docker containers. The Payara application server, PostgreSQL and the other runtime components run inside the container, so the solution is portable and works independently of the host operating system.

One shared Docker layer for two applications

The same Docker layer was built for both applications, in a way that left the business logic unchanged. This reduced both the development effort and later operating effort, and made the two systems consistent to manage.

Git-Nexus integration, next-next deployment

The containerised solution is built with Git and Nexus integration, turning deployment into a simple, step-by-step (next-next) process.

Externalised environment settings

The previously hardcoded environment settings - SP integration, mobile app integration, file server paths, server environment settings - were moved to an external setup file, where they can be changed easily.

4. Results

~5 mo	2	1	0
including testing	legacy Java applications	shared Docker layer	changes to business logic

Both applications moved to the new environment without a native migration. As a result of the project:

- **Fast deployment:** thanks to the Git-Nexus integration, deployment became a fast, next-next process
- **Environment independence:** the applications run in containers together with their full runtime environment, so they are portable and independent of the host operating system
- **Better performance:** the previously very slow applications perform better in the containerised environment
- **Unchanged business logic:** the proven business operation did not have to be touched, which reduced project risk
- **A consistent solution:** both applications are built on the same Docker layer
- **Configurable environment:** environment settings can be changed in the external setup file, without developer involvement

5. Key takeaways - what we recommend to other companies

The lessons from this project are especially relevant to organisations that need to move old but business-critical applications to a new environment:

- **A platform change does not have to mean a rewrite:** with containerisation, a legacy application can be moved to a new environment without taking on the full cost and risk of a native migration
- **One shared layer can serve several applications:** where runtimes are similar, a single well-designed Docker layer can cover the modernisation of several systems
- **Externalise configuration:** moving hardcoded environment settings to an external file brings a real operational benefit, even as a side effect of modernisation
- **AI tools help with legacy work too:** the AI-based tools used during development supported the team in a legacy environment too

About FrontX

FrontX Kft. is a professional IT project management and custom software development firm. Our ~20-person team builds on 20 years of enterprise experience — in financial services, insurance, and large-scale IT transformations. We help you realize your digital vision, from planning to go-live.

frontx.hu · info@frontx.hu · Behán László, Managing Director: +36 20 938 2448