![blankfactor]

How we helped a large payments processor step up their game

The client company we're featuring is one of the largest private and non-bank payment processors worldwide. This electronic payment processing company provides unified commerce solutions with over 530 payment methods in 150 currencies.

The challenge

The client company we're featuring is one of the largest private and non-bank payment processors worldwide. This electronic payment processing company provides unified commerce solutions with over 530 payment methods in 150 currencies.

The project

The company has over 20 teams of developers working on different projects and tech stacks. We partnered with them to help them scale their development efforts.

Blankfactor's expert architects, devs, and QA engineers work side-by-side on multiple projects, helping them reach new heights in the payments arena.

The project kicked off in August 2021 with ten devs, and it doubled in size in just a couple of months. The client found Blankfactor's expertise and help so valuable that they kept entrusting our partnership with more and larger teams of experts. We currently have over six ongoing and ever-growing projects.

Methodology

As part of the engagement plan with the client, Blankfactor's teams adopted the Agile + Waterfall framework the company is using for its development efforts. This was done to improve productivity and workflow visualization.

Nearshore

While the client's headquarters are based in North America, they have development centers across the globe. Their biggest one is located in Europe, and our developers in Sofia, Bulgaria, were able to work virtually side-by-side in the same time zone.
As quality is of paramount importance for the company, we're moving to a new stage that will lead the teams to a one-tester-per-developer ratio.

Some of the projects are:

A payment gateway

We're adding new functionalities to the company's main payment gateway. All transactions, such as credit card and alternative payments, go through this system layer. It processes around 5 million payment transactions a day. The tech stack is back end only with a .Net framework and MSSQL Server.

Building a REST API

We're writing a new version (2.0) of the company's current REST API. It will optimize all the existing functionalities, taking it to a new level.

Microservices

Finally, we can mention several ongoing projects in which we're rewriting existing applications with microservice architecture, dockers, and hosting in Kubernetes. The technologies include .NET Core 3.1 development platform, Kafka message broker, MSSQL database, Couchbase document store, and Redis cache-store.

Integration

We're working on a Getaway API that will serve as a mediator between the gateway processor and the operations triggered by the managing platforms. We're also integrating several products to easily manage client profiles and efficiently visualize metrics, chargebacks, and transactions, among others. For this project, we're implementing .NET Core, SOAP, MSSQL, MemSQL, GraphQL, ElasticSearch, MicroServices, and Docker.

Legacy technologies

The client encountered worrying performance issues derived from several of the legacy technologies they still use. Out of our engineering centers, we're helping them move their projects to a new architecture for billing processing and payouts for payment transactions using multiple technologies among which are .NET Framework 4.x development platform and jQuery.

The tech stack:

Microsoft .NET · Java · Microsoft SQL Server · kafka · MySQL · memSQL

GraphQL · elasticsearch · docker · jQuery · RabbitMQ
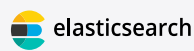
Results

Together, we have boosted performance to provide a seamless experience to the client's payment platform users.
We helped them improve their current architecture, allowing them to scale and update their systems according to market and clients' demands and needs.

Blankfactor has become their go-to partner for complex, ongoing software challenges for their core business. The partnership has been growing consistently since the relationship started.