

# simpel

## CASE STUDY: SIMPEL

Responsive and incident-free: Building a scalable web application starts with the right tools

### INDUSTRY

Mobile virtual network operator (MVNO) supplying SIM card services for large customer base

### APPLICATION

Self-service customer portal with back-end CRM, billing, telecom, CMS, and subscriber systems

### WEBSITE

www.simpel.nl

### CHALLENGES

- Deploy integrated, multi-supplier system with high availability
- Achieve efficient change management for releases
- Increase conversions from prospect to sales

### SOLUTION

Zend Server, Apigility, Zend Framework

### RESULTS

- Serving more than half a million customers, with enhanced features
- Zero incidents in first year of deployment
- Suppliers implement changes themselves

### ROGUE WAVE PARTNER



When your mission is to deliver fast, easy, and accessible SIM card subscriptions to a broad user base, simplicity is the name of the game. That means finding the right solutions from among a complex and cumbersome set of options.

[Simpel](#) is a low-cost mobile telephony company offering SIM-only services to more than half a million customers in the Netherlands. Originally a pure sales and marketing company, Simpel's focus on providing value and control to their customers drove the need to create a self-service website for subscriber functions. As demand and subscriptions grew, the website reached a tipping point, requiring a new solution that offered more capacity and the agility to respond to evolving customer needs. Enter [Enrise](#), a software development firm focused on bringing complex enterprise systems to the digital internet economy, to fundamentally shift the way things were done in the backend.

### Starting with the building blocks

In the beginning, Simpel evaluated a small number of mobile virtual network enablers (MVNE) to be sole suppliers of this new website and backend solution. Youri Treur, director e-commerce and support at Simpel, explains, "After two months, we concluded that there wasn't one single company that could supply our system. Each was good at one thing, such as telecom, but lacked specialization in web or customer service. We wanted to be able to scale up and sell other services on the same foundation, not just build telecom functionality." Once the decision was made to select multiple suppliers, each the best at what they do, the mission became a matter of coordinating different vendors, tools, and systems.

Kai Stevens, product owner at Enrise, recalls their first meeting. "We came in to build the front-end for the site, but what started as a few little blocks on the white board quickly grew into many, many blocks. We then learned what was actually needed, and immediately saw a big gap in the solution. Both the back-end and front-end were defined, but there was nothing in the middle that would keep things validated and synchronized."

This led to early overarching requirements. "It needed to be transparent to the end-user. The customer couldn't be aware that Simpel was distributing information over multiple systems. We also had to ensure that the right data went to the right systems and validated whether something was allowed or not. So the question became, how do we provide a solution to keep the entire data chain consistent without affecting users?"

## Queuing up the system

As planning continued, more requirements followed. Managing business rules required complex logic, and the various parts of the system needed consistent, easy to customize APIs to share information with each other. Achieving transparency for the end-user meant having to define clear system goals. Treur provides an example, “If a subscriber wanted to set a limit of not spending more than ten euros a month, the system had to create a trigger using defined workflows across all partners, like the conductor of an orchestra, while the front end had to be responsive and flexible.”

Thus was born the business logic layer (BLL), developed by Enrise to validate and handle all requests between the various systems to ensure everything was up to date and customers weren't slowed down. Says Enrise's Stevens, “Finding the solution was surprisingly easy. Based on our experience, we looked at the blocks on the white board and added the name Zend to each — there was almost a one-to-one match with many Zend features.”

Enrise implemented key BLL requirements using Zend solutions from Rogue Wave Software:

- Zend Server job queuing and job rules to coordinate data traffic between systems and manage business rules.
- Zend Server to automate deployment and accelerate changes to production servers.
- Zend Server Z-Ray to measure and log performance.
- Apigility to provide each supplier with their own BLL API in a consistent manner.

Stevens explains how they validated Zend Server. “We piloted it first, making a basic setup for job queuing by integrating the CRM and telecom suppliers. We also did performance testing on very specific technical things, like inserting two thousand new contracts within a time period and measuring performance and concurrency between systems. We tested Zend Server, pushed it to the limit, and it just worked.”

Implementing business logic was equally straightforward. “We used the Zend Server job rules module to manage resources. For example, when a new billing cycle starts, a lot has to change. We started with manual changes, but now, job rules check which customers need to start a new cycle only during minimal load times on the system, keeping everything efficient.”

## Changing the game

Finding the right path from complexity to simplicity can be challenging, Sempel's Treur sums up the project's philosophy to achieve this: “Being a game changer is all about time to market. This means having efficient change management, being flexible as a company, providing good self-service, while also having good conversion from prospect to sales. Finding suppliers that are the best at what they do helped us do this.”

Stevens offers his own insights as well, "When you have a large unknown, you can't ignore it or tackle it. You must make it smaller and look for strategies to split it into smaller pieces. We went from one supplier doing everything to several, and gave them the ability to perform changes themselves. With the new solution, release cycle times have greatly decreased, providing a shorter time to market for Simpel."

"We're now serving over half a million customers on a brand-new stack and with many benefits. Zend Server job queuing is really, really, important to this. It's been up for a year now with zero incidents."



[www.enrise.com](http://www.enrise.com)

With our focus on technology, Enrise combines the stability of existing software systems with the dynamics of the internet. We help transform organizations to digital first while maintaining stability and flexibility.



[www.roguewave.com](http://www.roguewave.com)

© 2017 Rogue Wave Software, Inc. All Rights Reserved

Rogue Wave Software is the largest independent provider of cross-platform software development tools, components, and platforms in the world. Through decades of solving the most complex problems across financial services, telecommunications, healthcare, government, academia, and other industries, Rogue Wave tools, components, platforms, and services enable developers to write better code, faster.