## QUOTE CREATION MADE EASY WITH LIGHTNING WEB COMPONENTS

## **Case Study of Heizungsmacher AG**

The Salesforce Spring '19 release announcement was very exciting for everyone in the Salesforce developers community. It introduced **Lightning Web Components (LWCs)**, a new programming model for Lightning component development, based on modern web technologies. A few months have passed since the release, but still not many users have had the opportunity to work with LWC's. We had – and we are thrilled with the results!

Let's start with a bit of backstory. A year ago we developed a **CPQ-like functionality** in Salesforce for one of our clients: **Heizungsmacher AG**, a company which specializes in planning and coordination of renewable energy installations like heat pumps or solar roofs for its customers in Switzerland. The most important value which Heizungsmacher AG gives their customers is **full transparency** – they receive a clear and very detailed offer. Such quotes are very complex and, although created from predefined bundles, final offers vary from each other and have to be very household-specific.

The system we created was quite complex and needed hours of configuration: over 360 products, dozens of product bundles, hundreds of dependencies between products and rules based on each household's characteristics. This investment, however, **made quote creation process easy and enjoyable**. There was one issue left, though: the **Quote Editor** – a front-end application for editing quote's details, which stood on top of this complex logic and data model. It was fully functional, but well... unfortunately came out to be very slow. Users weren't very eager to use it and even found a new name for the **Drag&drop functionality** for changing quote line order: "Drag&wait". This is what users saw most of the time while using the Quote Editor:





Back in 2014, Salesforce came up with its own programming model for building Lightning components called **Aura**, which in 2019 seems to be a bit outdated in comparison to standards that have emerged in the meantime. A lot has changed since then in the technology landscape – **JavaScript** has become a core technology behind the internet. When we were building the Quote Editor, however, we had no other choice but to use Aura.

That's why the announcement of the LWCs was so big. It enabled developers to build on Salesforce with the modern JavaScript, using core Web standards. It basically means that LWCs are **extremely fast, lightweight and deliver exceptional performance**. Lightning fast finally became a thing!

It doesn't mean that Aura will from now on be replaced by LWCs – it will still be supported and both components built on both frameworks (Aura & LWCs) can coexist in the same application. Nonetheless, you should definitely consider designing LWCs from now on if:

- you are developing brand new custom Lightning Components that have not been built yet,
- you need better performance of existing apps with heavy Aura-based components you may consider rewriting them entirely or go for a hybrid approach and rewrite only its child components.

LWCs were definitely something that we needed in order to improve app speed and usability. We created the brand new **Quote Editor 2.0** which not only had to work faster but also had to be much more complex than the previous one to enable users to create quotes more quickly and flexibly.

## What are the advantages of employing this new JavaScript-based component?

- Full inline editing for all the quote lines in one view
- **Immediate response** when adding, deleting and editing quote lines (with instant price calculations based on the price book!)
- Instant saving, cancelling, undoing and redoing of all the changes.

No more spinning wheel! We've created a **flexible component** that can (and will) be also used in the future for editing project details after sales, on objects other than Quote.

