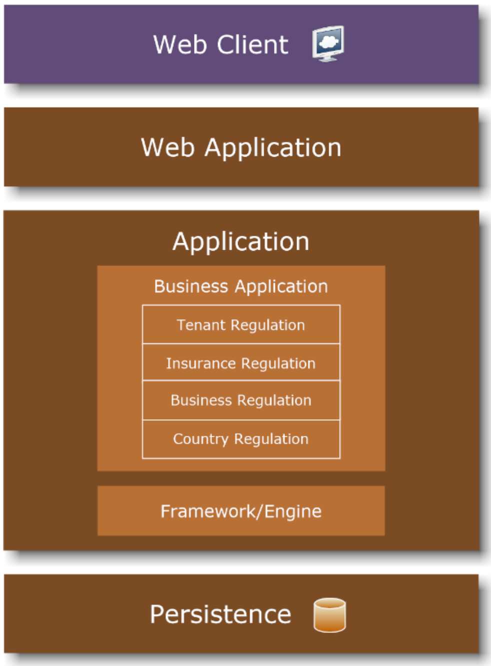
Payroll Engine - Design of a scalable payroll software



Driven by bureaucratization, digitization and constant technology change, the design of modern software is a complex challenge. Based on practical experience, the scalability of business cases is particularly crucial for payroll software.

To reduce complexity, software systems are usually divided into different areas (layers or services). These are usually the front end/client with the user interface (UI) and the back end/server, consisting of the application and the data persistence (database). From a value creation perspective, the application is the software core, as it implements the business processes. The user interface and persistence should ideally be generated from the business model.

If the application is monolithic, any customization must be done by highly skilled software engineers who do not always know the relevant business requirements.

When designing the Payroll Engine, we followed the concept of *Domain-Specific Application Frameworks* [1)] (business-oriented application frameworks), which provides for the separation of the application into a framework (framework) and the domain/business application. The framework contains the basic functionality and is developed by software engineers who are only involved in fundamental changes. The domain/business application is developed by domain specialists who are provided with appropriate tools for this purpose. The specialists are expected to have a certain technical understanding.

This separation reduces the administrative, financial and communicative effort between business and framework developers. A central design aspect is the correct distribution of knowledge, which enables agile development of business applications and reduces the transfer of know-how between the parties, e.g., in the event of personnel changes, to what is necessary.

In the Payroll Engine, the REST API represents the framework and the regulations a part of the business application. The regulations include the business cases, the payroll calculation and the evaluations/reports.

Another challenge in designing a payroll software is that the business functionality comes from different sources, such as collective bargaining agreements, insurance policies, etc. The Payroll Engine integrates each business source as a separate Regulation and merges them in a layered model into a business application. Analogous to the image layer in an image processing program, all rule layers result in the overall image of the business application. The adjustments for the accounting client are integrated into the top rule layer (image layer).

1) Mohamed E. Fayad & Ralph E. Johnson – Domain-Specific Application Frameworks
ISBN: 978-0-471-33280-0, Wiley