FenixEdu Common Architecture

Introduction

The FenixEdu Common Architecture is how most of the development made in by our FenixEdu team, and for the project as a whole. It comprises the technologies that we use and also the architecture we put in place so you can extend our applications for your organisation needs.

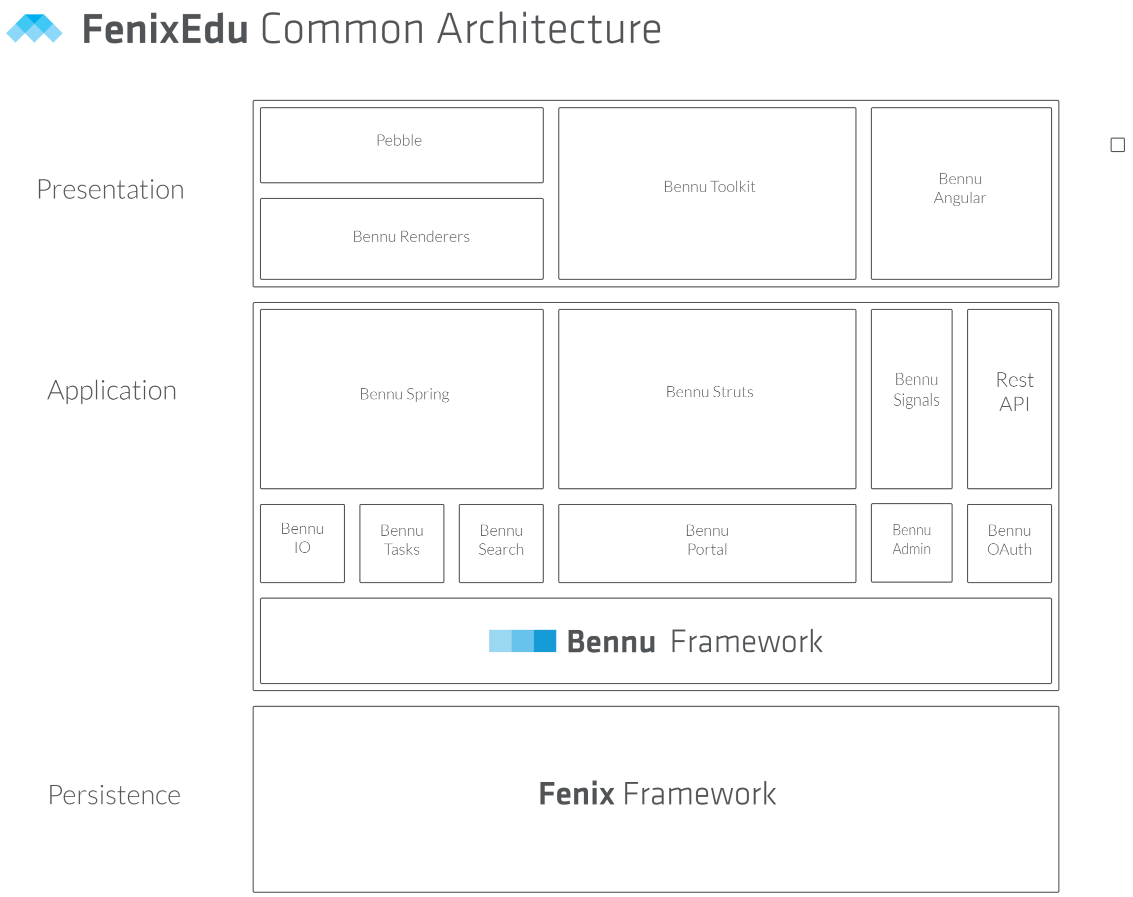
Technologies

Most applications have a great number of dependencies and uses several technologies that have been added to the project along with its development. Instead of talking about all of them we will just mention some of the more important ones to our architecture and its use will be described has we explain our architecture.

- Java 8
- Java Server Pages
- Java Server Faces
- Struts
- Maven
- MySQL
- Fenix Framework
- Bennu
- Bennu Renders

Application Structure

This is how our architecture looks like:



Persistence Layer

Fenix Framework is a Java ORM that enables you to express the domain model as entities, the relations between the entities and its attributes. After having this specification, the framework automatically creates the Classes that implement the existing relations. You just have extend that classes and implement the application logic.

The framework takes care of the persistency layer, saving the changes to the entities on the database (MySQL). It also handles the transactions and provides ways to ensure consistency. All the data is kept on memory and the transactions are also done on memory. The updates done are saved on the database from time to time. To learn more about this take a look at <MISSING>

Application Layer

For the applicational layer we have developed a framework that provides a bunch of common artefacts consistently across our applications while also working as a bridge between the Fenix Framework and the application. Its called Bennu Framework.

Bennu comes with support for cron tasks, adaptive file storage, indexing, user and menu management. This allows us to focus on the application core and leave the menial work for the framework.

Bennu also allows you to combine several application frameworks. We currently provide adapters for two, Spring and Struts. This makes it possible to have code develop in many different technologies and yet provide a consistent user experience across those frameworks. It is even possible to combine JVM code from other languages like Python, PHP or Closure into Bennu.

You can learn more all about it here.

Presentation Layer

Modularisation