

# Softwarearchitektur

Software Engineering II



***Optima Connect***

**Projekttitle: BonoboBoard**

Jakob Hutschenreiter (1419081)

Jiesen Wang (9839152)

Nick Kramer (3122448)

Patrick Küsters (2598689)

Peter Moritz Hinkel (2783930)

DHBW Mannheim

25. Februar 2022



## Änderungshistorie

Revision	Datum	Autor(en)	Beschreibung
1.0	22.02.2022	NK PK JW PH JH	A: 1, 2, 3
1.1	25.02.2022	NK PK JW PH JH	C: 2, 3

Abkürzungen: Hinzugefügt/Added (A), Änderung/Changed (C), Löschung/Deleted (D)

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Architektur</b>	<b>1</b>
<b>3</b>	<b>Zusammenfassung</b>	<b>2</b>



## 1 Einleitung

Dieses Dokument befasst sich mit der Softwarearchitektur des Projekts „BonoboBoard“. Es wird auf die wesentlichen Komponenten eingegangen, wie diese miteinander interagieren und in welcher Form sie von dem verwendeten Web-Framework „Django“ vorgegeben werden.

## 2 Architektur

Bei dem BonoboBoard handelt es sich um eine Webanwendung (Client-Server), die mithilfe des Django Frameworks erstellt wurde. Gehostet wird die Serveranwendung in einem Docker Container auf einem VServer. In diesem Container befindet sich ein nginx Server, der Anfragen an einen gunicorn Server weiterleitet. Dieser gunicorn Server beinhaltet die eigentliche Django Anwendung (siehe Abbildung 1).

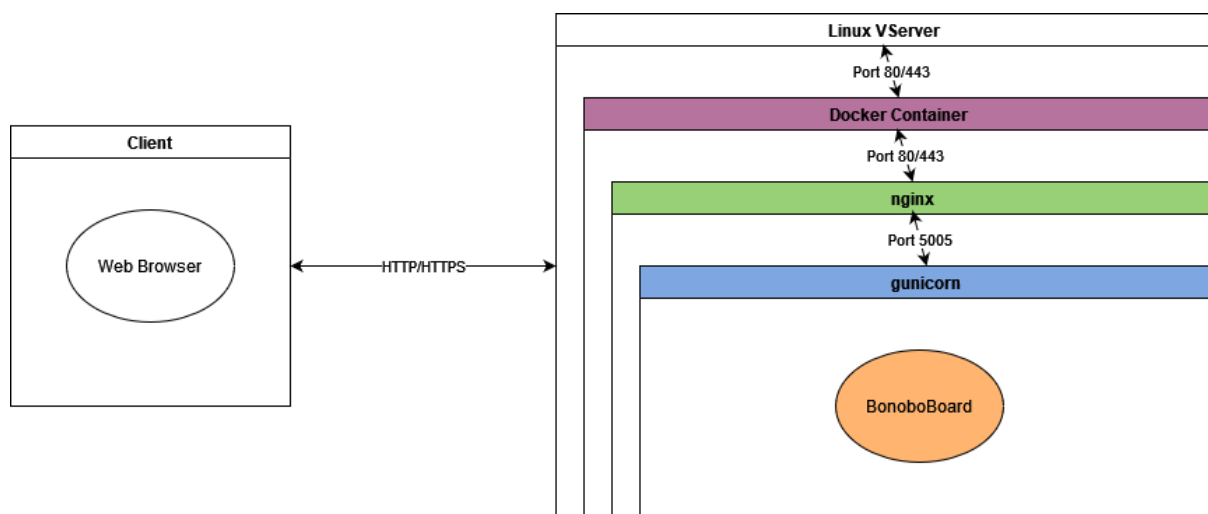


Abbildung 1: Serverstruktur BonoboBoard

Die Anwendung selbst folgt dem „Model-View-Presenter“ Schema (MVP), welches aus dem „Model-View-Controller“ Schema (MVC) hervorgegangen ist. Dieses wird vom Django Framework vorgegeben.

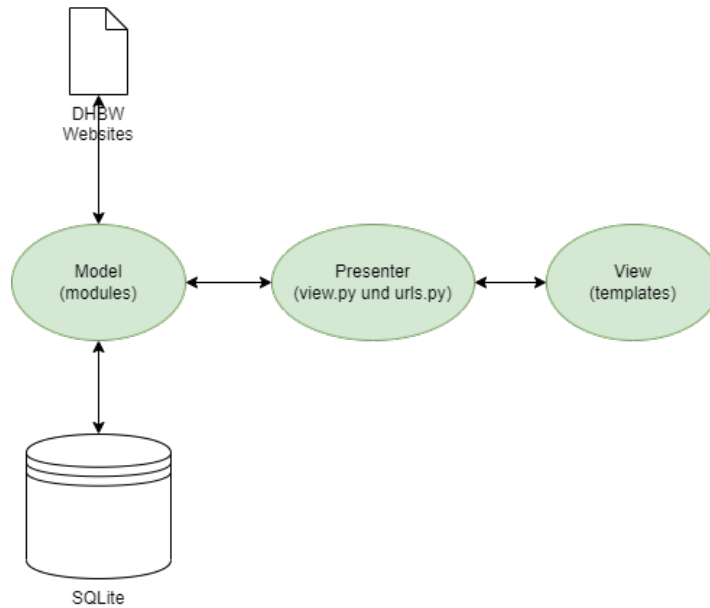


Abbildung 2: Model-View-Presenter Schema des BonoboBoards

Abbildung 2 zeigt das Model-View-Presenter Schema des BonoboBoards und wie die einzelnen Komponenten miteinander agieren. Diese Komponenten werden im Folgenden näher beschrieben.

**Model** Das Model beinhaltet die Daten, welche angezeigt werden sollen. Es stellt die Logik der View dar und wird vom Presenter gesteuert. Das Modell ist die Schnittstelle zu den Scrapern, die die Daten von den jeweiligen Websites beziehen und zu einer SQLite Datenbank, welche aufbereitete Daten hält. Das Model wird durch Objekte in `modules/dhbw` realisiert, über die es mit der Datenbasis interagieren kann.

**View** Die View ist für die Darstellung der Website gegenüber den Nutzer\*innen verantwortlich. Sie hat keinen Zugriff auf das Model oder auf den Presenter, sondern kümmert sich lediglich darum, wie die bereitgestellten Informationen angezeigt werden und nimmt Nutzereingaben für den Presenter entgegen. In Django werden die Views in Form von HTML-Templates realisiert. Dabei handelt es sich um HTML-Dateien, bei denen variable Informationen mittels spezieller Tags gekennzeichnet werden.

**Presenter** Der Presenter ist das Bindeglied zwischen View und Model. Er steuert, welche Elemente aus dem Modell an die View übergeben werden, damit diese angezeigt werden können. Der Presenter ist für die Verarbeitung von Nutzereingaben zuständig. Diese Funktion wird in Django von zwei Modulen übernommen: `urls.py` und den `urls.py` Modulen. `urls.py` ist für die Weiterleitung von Nutzeranfragen an die entsprechende Funktion in `view.py` zuständig, während `views.py` die Anfrage entgegennimmt und ein entsprechendes HTML-Template mit Informationen befüllt.

### 3 Zusammenfassung

Bei dem BonoboBoard handelt es sich um eine Client-Server-Anwendung, die ein Framework zur Grundlage hat, dass weitere Teile der Architektur vorgibt. Das verwendete Framework Django folgt

dem Model-View-Presenter Schema und wird verwendet, um Informationen aus einer Datenbank oder - mittels Scrapern - von Websites zu beziehen und anzuzeigen.