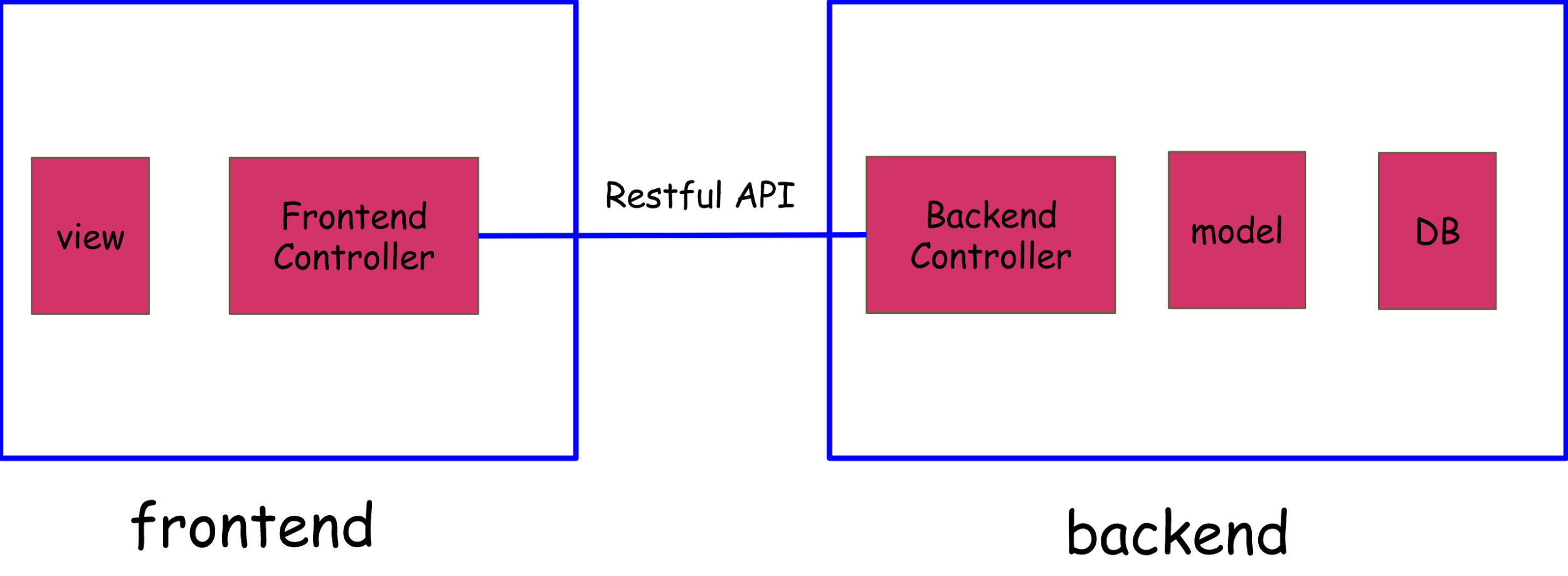
**Design Documents**

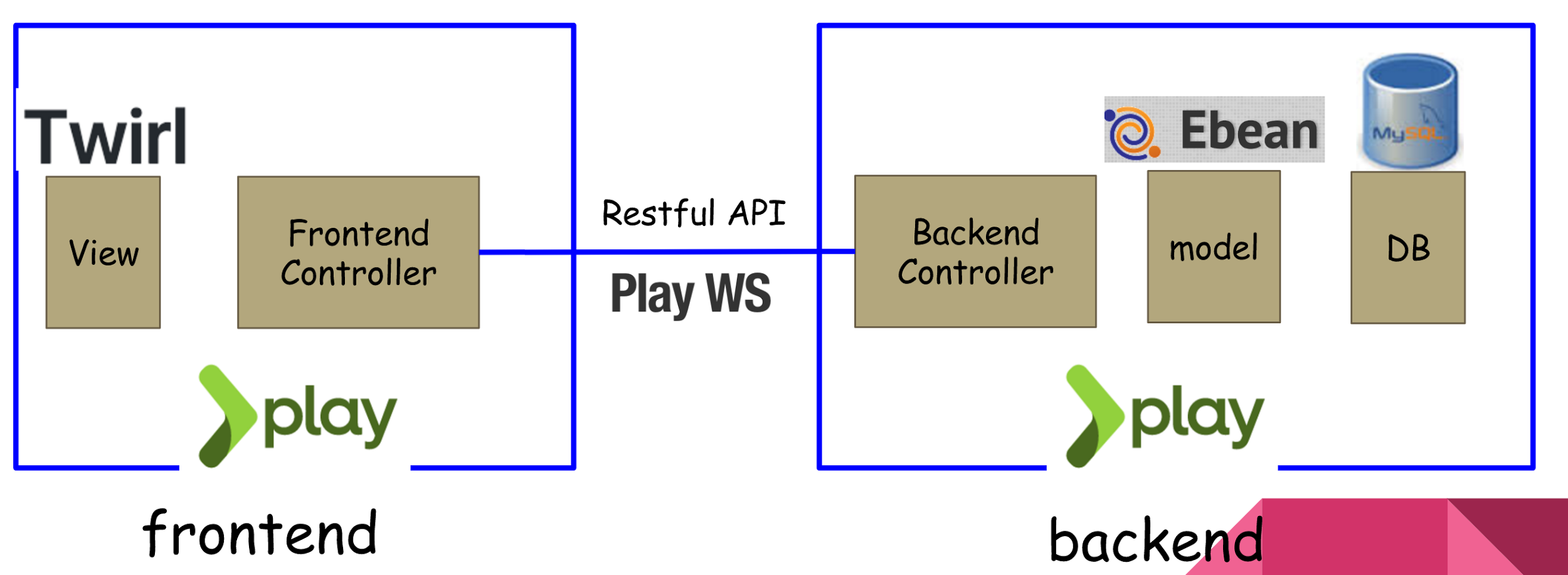
1. Overall system design

The system design is shown as Figure 1. In this project we divide the whole project into two parts: frontend and backend, so that some people can focus on frontend/backend without worrying about backend/frontend things. What they need to do is to invoke the services provided in order to implement a whole functionality. In the frontend, there are view and frontend controller. View is how we display the website while frontend controller is to gather data using restful api from backend. In the backend, there are backend controller, model and database. Backend controller can respond to the requests from frontend. Model communicates with database.

In the system we use two play framework. In the frontend, we use Twirl template engine for view, which is a powerful scala-based template engine. For frontend controller we use play, written in Java. We use Play WS library to send/respond to http request to implement restful api. For backend controller we use play framework. The database we use is relational database, because all the required field is decided. Here we use MySQL as database. We use Ebean ORM to manipulate the database operation with MySQL database. We use sbt to manipulate dependencies.



**Figure 1. Overall system design**



**Figure 2. Overall system implement**

1. Database schema design

The relation schema of this system is as Figure 3. We use

User - store the login information of registered user

Profile - store the detailed information of a user

ConferenceDetail - store the detailed information of a conference

Criteria - review criteria of a certain reviewer to a certain paper

Question - review question of a certain reviewer to a certain paper

Conference - store the relation information of a user and its conference

Review - store the relation information of a reviewer to a paper

Paper - store the information of a paper(including user who submit it, conference it belongs to, etc)

