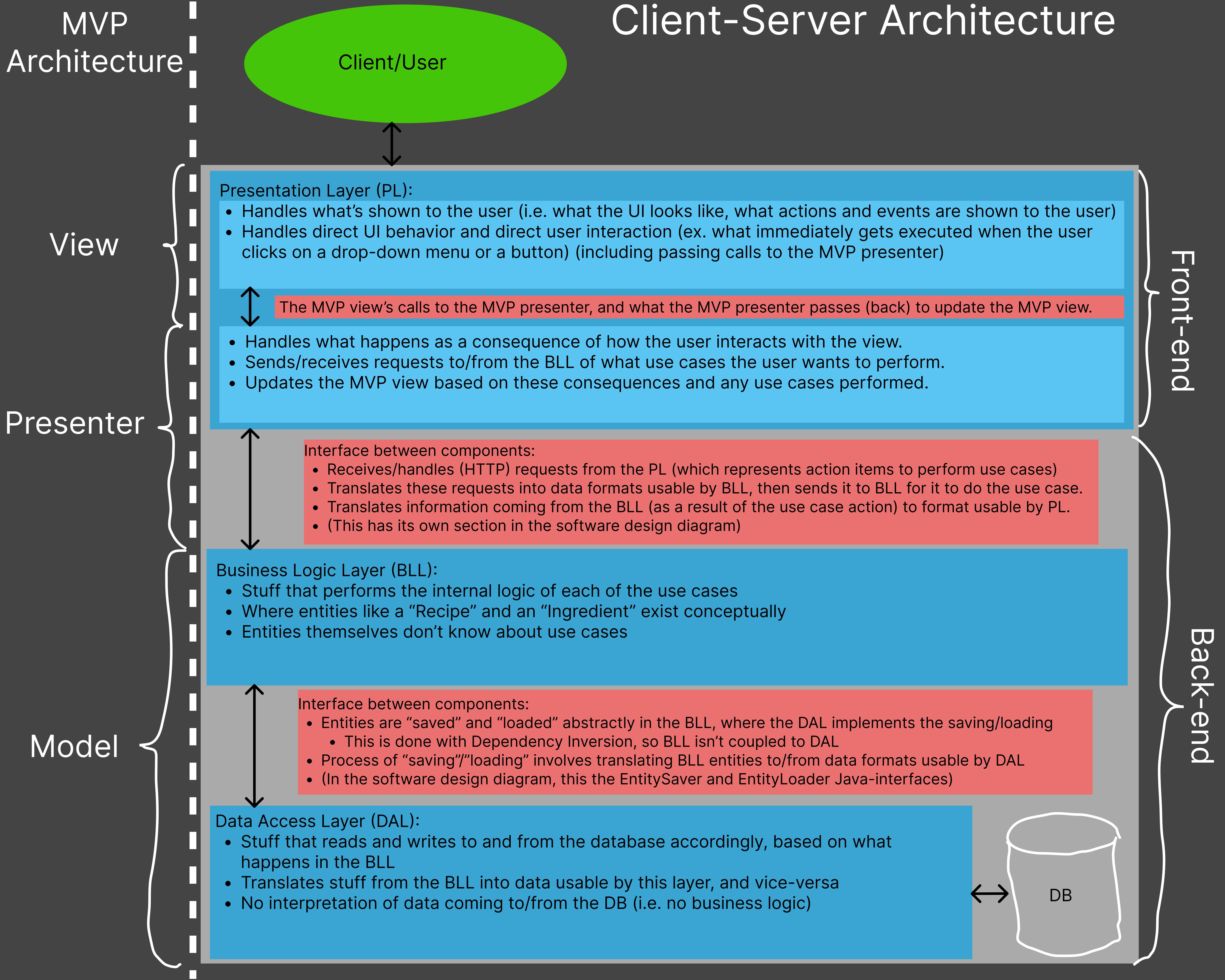
Client-Server Architecture

MVP Architecture



Presentation Layer (PL):

- Handles what's shown to the user (i.e. what the UI looks like, what actions and events are shown to the user)
- Handles direct UI behavior and direct user interaction (ex. what immediately gets executed when the user clicks on a drop-down menu or a button) (including passing calls to the MVP presenter)

The MVP view's calls to the MVP presenter, and what the MVP presenter passes (back) to update the MVP view.

- Handles what happens as a consequence of how the user interacts with the view.
- Sends/receives requests to/from the BLL of what use cases the user wants to perform.
- Updates the MVP view based on these consequences and any use cases performed.

Interface between components:

- Receives/handles (HTTP) requests from the PL (which represents action items to perform use cases)
- Translates these requests into data formats usable by BLL, then sends it to BLL for it to do the use case.
- Translates information coming from the BLL (as a result of the use case action) to format usable by PL.
- (This has its own section in the software design diagram)

Business Logic Layer (BLL):

- Stuff that performs the internal logic of each of the use cases
- Where entities like a "Recipe" and an "Ingredient" exist conceptually
- Entities themselves don't know about use cases

Interface between components:

- Entities are "saved" and "loaded" abstractly in the BLL, where the DAL implements the saving/loading
  - This is done with Dependency Inversion, so BLL isn't coupled to DAL
- Process of "saving"/"loading" involves translating BLL entities to/from data formats usable by DAL
- (In the software design diagram, this the EntitySaver and EntityLoader Java-interfaces)

Data Access Layer (DAL):

- Stuff that reads and writes to and from the database accordingly, based on what happens in the BLL
- Translates stuff from the BLL into data usable by this layer, and vice-versa
- No interpretation of data coming to/from the DB (i.e. no business logic)