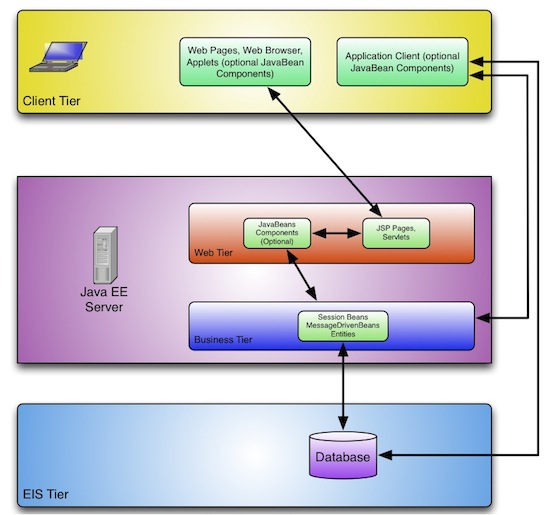
**Architectural Patterns and styles**

Model-View-Control architecture (MVC)

Adhering to the MVC design pattern provides us with numerous benefits:

* **Separation of design concerns:** Because of the decoupling of presentation, control, and data persistence and behaviour, the application becomes more flexible; modifications to one component have minimal impact on other components.
* **More easily maintainable and extensible:** Good structure can reduce code complexity. As such, code duplication is minimized.
* **Promotes division of labour:** Developers with different skill sets are able to focus on their core skills and collaborate through clearly defined interfaces



**View (Client tier)**

This tier runs on the client system and encapsulates the various components that a client system may use to access the Java EE server-side tiers. These components include dynamic web pages, Java applications and Java applets …

**Controller (Java EE server)**

 The middle tier's business functions handle client requests and process application data, storing it in a permanent data store in the data tier.

1. **Web tier**

The web tier consists of components that handle the interaction between clients and the business tier.

1. **Business tier**

The business tier consists of components that provide the business logic for an application. Business logic is code that provides functionality to a particular business domain…

**Model (EIS tier)**

The EIS tier consists of database servers, enterprise resource planning systems, and other legacy data sources. These resources typically are located on a separate machine than the Java EE server, and are accessed by components on the business tier.

Dependency injection

Creates new objects using dependencies, decouples code, makes it cleaner, easier to modify and easier to reuse.

Blackboard pattern

To archive scalability, blackboard multiple processes to work closer together on separate threads, introduction of this pattern will help out multiple process of the buzz system to run efficiently as the pattern emphasizes multiple processes working together

Layered Architecture

System will be separated through layers; there will be User Interface layer, services layer and process layer which include Business logic and data. User Interface layer will handle interaction like receiving input from users, the service layer will provide the human layer with services like opening a buzz space and commenting on the buzz thread and lastly process layer will process services rendered for authorization and quality check like plagiarism. Separation through layers will enhance performance, manageability and reusability.

*Client/Server*

**For communication of the server which is buzz system with users, this pattern have benefits of security as all data will be stored on the buzz system server and ease of maintenance as server is responsible of repair with client knowing of damage.**