**High-Level Details**

* The **Pipes and Filters** architectural pattern is one of the **Messaging Patterns** that help in splitting a large set of operations on a message into various processes so that each process can work on the message independently and complete the transformation of the message.
* Here each process will be called “**Filters**” and they are connected using the channels or connectors called “**Pipes**”.
* All the filters implement a common interface so that all will conform to the contract that they are supposed to work on.
* Message from the source will be set as input to the process and the output or result of the process will be sent as an input to another process and so on until it reaches the sink or destination process. (**Note:** Here the terms “**Process**” and “**Filters**” are used interchangeably).
* In the preceding diagram the first process sets the message priority by taking the incoming message using a pipe and then sends the output to another process to encrypt the message.
* Then the outgoing message is published to the Queue.

**MVC**

Separates presentation and interaction from the system data. The system is structured into three logical components that interact with each other. The Model component manages the system data and associated operations on that data. The View component defines and manages how the data is presented to the user. The Controller component manages user interaction (e.g., key presses, mouse clicks, etc.) and passes these interactions to the View and the Model.