Super Rummy Architecture

An **event-driven system architecture** is used for Super Rummy.

**Description:** In this design pattern, there exists a state-based controller with several components under its control. Events are sent from the components to the controller, and the controller responds with instructions. In our case, we have named these events “actions” and the instructions “events.” The server will receive an “action,” such as a request to move a card from point A to point B, and respond by sending an “event” to the proper component to perform the request after verifying it. These “events” will also notify all clients of resulting changes.

**Justification:** We chose this design pattern because we needed an intuitive way to prevent cheating. With event-driven system architecture, the state based controller, or server, in our case, will be able to prevent illegal moves because it will be able to process the events sent to it before sending instructions back.

