Week 3: REST Services (Practise Assignment)

Following points highlights Trello's confirmation to REST Design Principles and Constraints:

1) **Client-server :**
The web application(client) sends requests to server, and in response, server provides meaningful response messages which client can consume and process accordingly. The client is the trello web application and the server is the backend that responds to client responses.

2) **Layered :**
Cannot actually prove that Trello is layered, but it can be assumed that it has a client-server layered architecture. This has helped Trello to achieve Separation of Concerns and modularity.

3) **Interactions are stateless :**
API requests include the information to grab all resources so that the server does not have to save information about user states. A single request from Trello UI is independent and is self-contained. Trello Client sends cookies along with the equests as authentication information.

4) **Clients can cache responses – cacheable :**
Web browser saves login as cookies so that can stay logged in to make requests. HTTP Response sent from server contains cache-control headers with different values of max-age.

5) **Uniform interface for communication**
Done through URLs and data communicated using XML or JSON. Requests are sent using common HTTP methods like GET, POST, PUT and DELETE, the resources with URI's. The response bodies are written in JSON format. This format is consistent across the application.