Designing a distributed video game

Consider a distributed video game that manages virtual worlds where users are represented using avatars that have specific roles. Roles define the properties of an avatar and the type of actions that it can perform (i.e., super jump, extra life bar, invisibility, resurrection). Also consider that the video game is implemented via a set of servers distributed over different geographical regions (i.e., America, Europe), where each server manages a specific aspect of the game (i.e., the global state of the game, the decoration that a user sees when he/she accesses a virtual world). In consequence, servers have to collaborate for:

- Notifying the connection/disconnection of a user and notifying his/her last state.
- Broadcasting the change in the state of an avatar as a result of some action (i.e., it changes its state from invisible to visible, it enters a new virtual region in the world).
- Receiving/Notifying information about the actions of an avatar (an avatar shoots and hits another avatar).

TO DO

- Define your hypothesis concerning the game: performance requirements with respect to given actions, security of some data, dependency between functions and data necessary to perform them, whether specific data are critical (i.e., they must be available continuously, they must be updated and visible for all participants).
- Describe the number of servers that compose your video game and their roles. In particular describe how data is organized (e.g., user profiles, global state, decoration).
- Which is the architectural model of your game? Explain.
- Give examples of code that implement the administration of the servers.
- Characterize your servers with respect to the state management, the communication model and the execution model they implement. Justify.