1)

Le principal avantage de cette architecture c’est qu’étant donné que tous les calculs sont effectués coté serveur, il est impossible pour les joueurs de tricher. En effet si les calculs étaient faits du côté client pour chaque joueur, on peut imaginer un hackeur modifiant les valeurs dans sa mémoire pour envoyer de mauvaises données aux serveurs. Le principal inconvénient de cette architecture est la bande passante. Par exemple, si le jeu est un jeu en temps réel, le client devra envoyer des données au serveur à chaque étape, et le serveur devra envoyer les données à tous les autres joueurs. Il devient aussi vite complexe d’implémenter cette architecture sur des jeux plus complexes, sans souffrir de problème de désynchronisation. Une gigue élevée peut également provoquer un décalage et rendre le jeu moins réactif. Un taux de perte élevé peut quant à lui rendre le jeu instable et la communication avec les autres joueurs difficile. Et un délai élevé peut entraîner un décalage entre ce que le joueur voit et ce qu’il se passe concrètement sur le serveur, ce qui peut être frustrant.

2)

L'utilisation de TCP pour les jeux en ligne peut être souhaitable dans certains cas par exemple pour des jeux en tour pas tour, mais elle n'est pas obligatoire. On peut utiliser UDP à la place, ce qui peut offrir une expérience de jeu plus réactive. Le protocole TCP exigeant un accusé de réception pour chaque paquet reçu, des retards sont possibles si des paquets sont perdus ou retardés. Pour les mêmes raison UDP est utilisé pour le Cloud Gaming. Comme UDP est un protocole sans connexion, il peut être utilisé pour les jeux en temps réel où les actions doivent être effectuées immédiatement, sans attendre la confirmation de l'autre partie.