Q: Explain why transient synchronous communication has inherent scalability problems, and how these could be solved.
A: The problem is the limited geographical scalability. Because synchronous communication requires that the caller is blocked until its message is received, it may take a long time before a caller can continue when the receiver is far away. The only way to solve this problem is to design the calling application so that it has other useful work to do while communication takes place, effectively establishing a form of asynchronous communication.

Q: Consider DNS. To refer to a node *N* in a subdomain implemented as a different zone than the current domain, a name server for that zone needs to be specified. Is it always necessary to include a resource record for that server's address, or is it sometimes sufficient to provide only its domain name?
A: When the name server is represented by a node *NS* in a domain other than the one in which *N* is contained, it is enough to give only its domain name. In that case, the name can be looked up by a separate DNS query. This is not possible when *NS* lies in the same subdomain as *N*, for in that case, you would need to contact the name server to find out its address.

Explain in your own words what the main reason is for actually considering weak consistency models.
A: Weak consistency models come from the need to replicate for performance. However, efficient replication can be done only if we can avoid global synchronizations, which, in turn, can be achieved by loosening consistency constraints.

Linearizability assumes the existence of a global clock. However, with strict consistency we showed that such an assumption is not realistic for most distributed systems. Can linearizability be implemented for physically distributed data stores?
A: Yes. Linearizability assumes loosely synchronized clocks, that is, it assumes that several events may happen within the same time slot. Those events need to be ranked adhering to sequential consistency.

Consider a personal mailbox for a mobile user, implemented as part of a wide-area distributed database. What kind of client-centric consistency would be most appropriate?
A: All of them, actually. What it boils down to is that the owner should always see the same mailbox, no matter whether he is reading or updating it. In fact, the simplest implementation for such a mailbox may well be that of a primary-based local-write protocol, where the primary is always located on the user's mobile computer.