

Instantiate Architectural Elements, Allocate Responsibilities, and Define Interfaces

Design Decisions and Location	Rational
Request-queue strategies for application servers by using a singleton pattern	RequestQueue is a frequent method to manage application server failovers. However, the singleton pattern is an architectural design pattern that limits the instantiation of a class to one object. For example, sending many requests to the DB or the server which makes a single connection for each request makes this costly for us. But by using a singleton pattern, we make one connection for all user requests.
Multiple-queue implementation	<p>The Multiple-queue method is another solution for keeping our server from failovers. This method requires a single pending queue and an active queue for each application server.</p> <p>With separate queues,</p> <ol style="list-style-type: none">1) No additional data needs to be stored.2) Any record assigned to the server's queue will stay active.3) In case of server failovers and outages, the server's queue will be added to the pending queue.