A suitable Java class to use for the holding area in the alternative version of TopicServer described would be a BlockingQueue from the java.util.concurrent package.

A BlockingQueue is a thread-safe queue that supports operations such as put, which adds an element to the end of the queue, and take, which retrieves and removes the first element of the queue, blocking until an element is available if the queue is empty. This makes it an appropriate choice for storing the messages in the holding area, as it ensures that the retrieval of messages by the separate thread and their subsequent delivery to subscribers is done in a synchronized and thread-safe manner.

The pattern of synchronization provided by the BlockingQueue is one of mutual exclusion, where only one thread can access the queue at a time, and the put and take operations are atomic, meaning that they are indivisible and cannot be interrupted by other threads. This ensures that messages are not lost or duplicated, and that they are delivered to subscribers in the order they were received. The blocking behavior of the take method also ensures that the separate thread waits until a message is available before attempting to deliver it, avoiding unnecessary CPU cycles and improving performance.