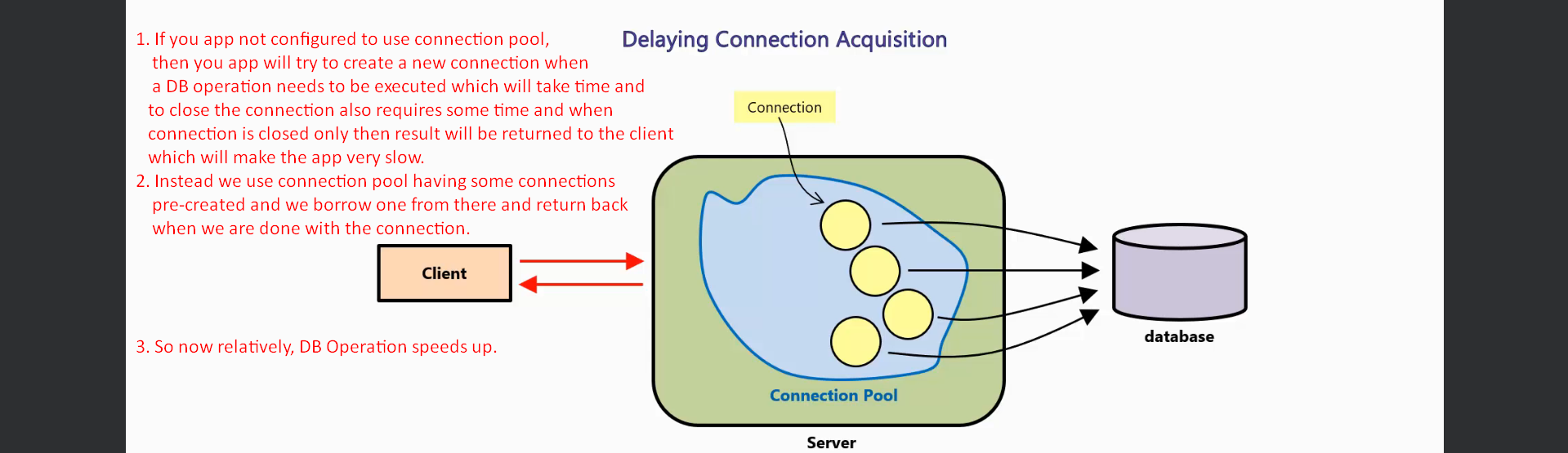
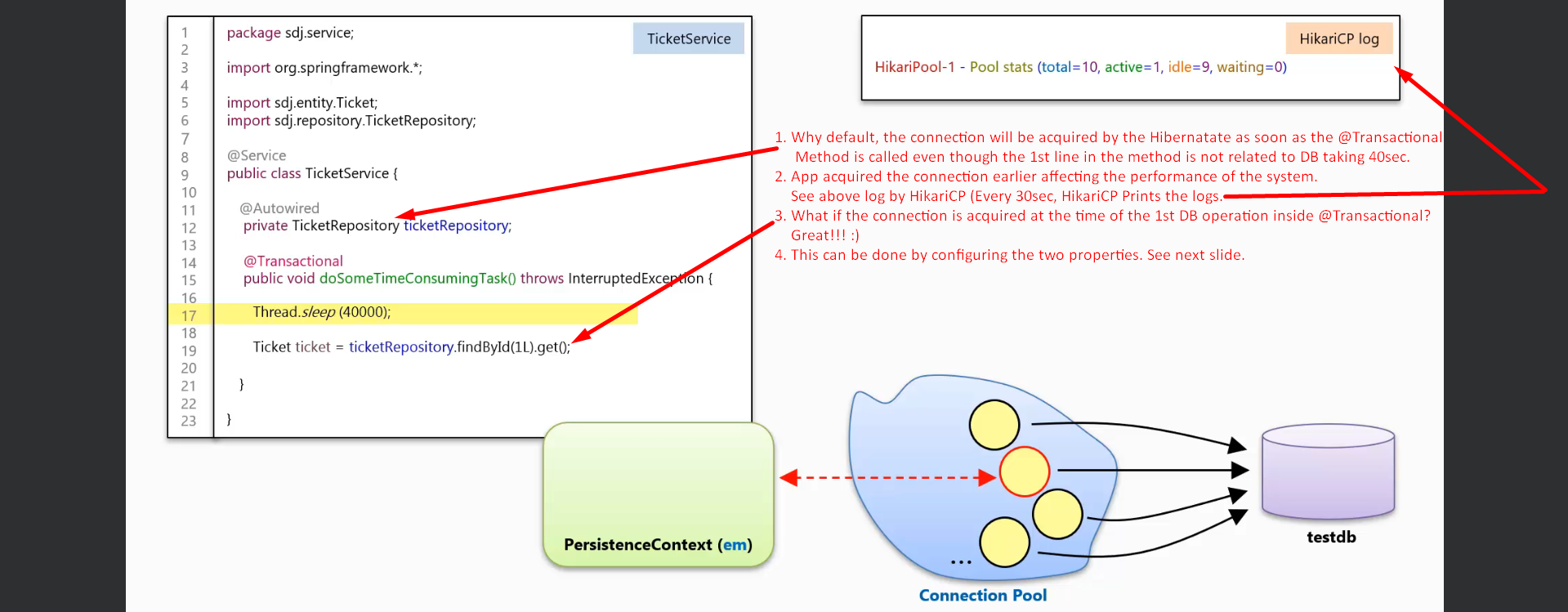
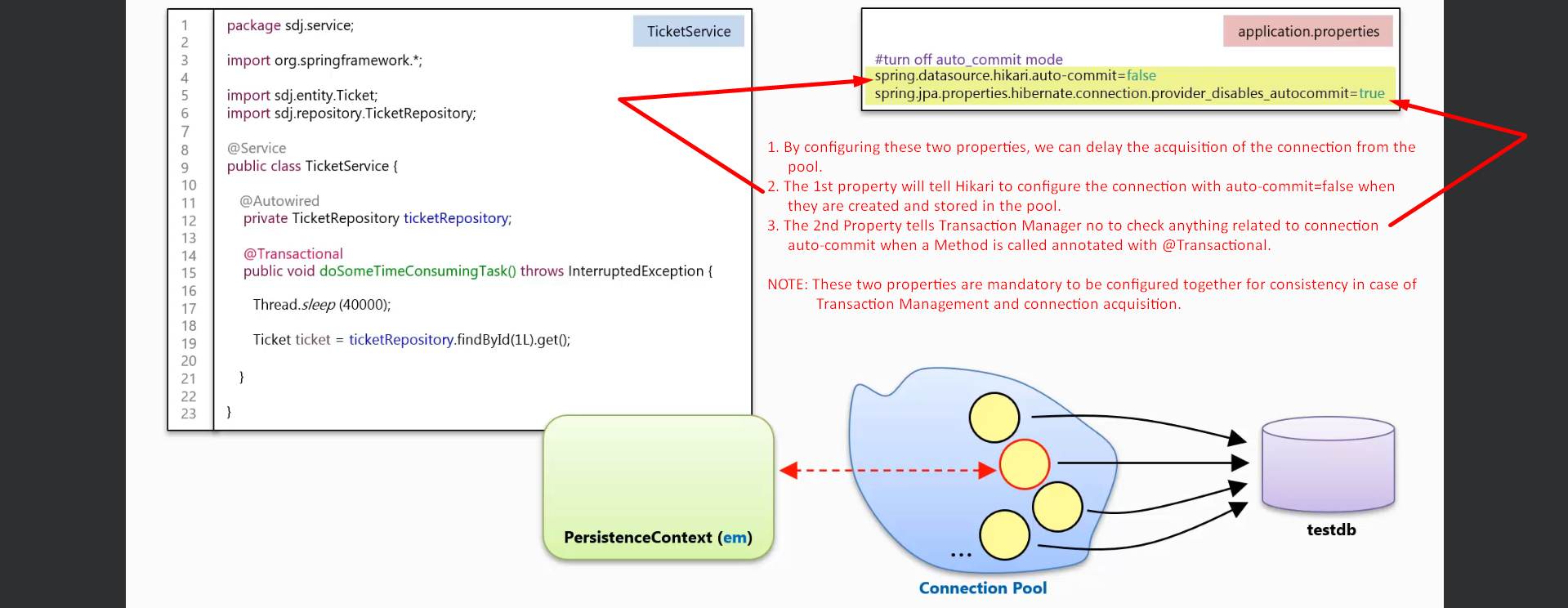
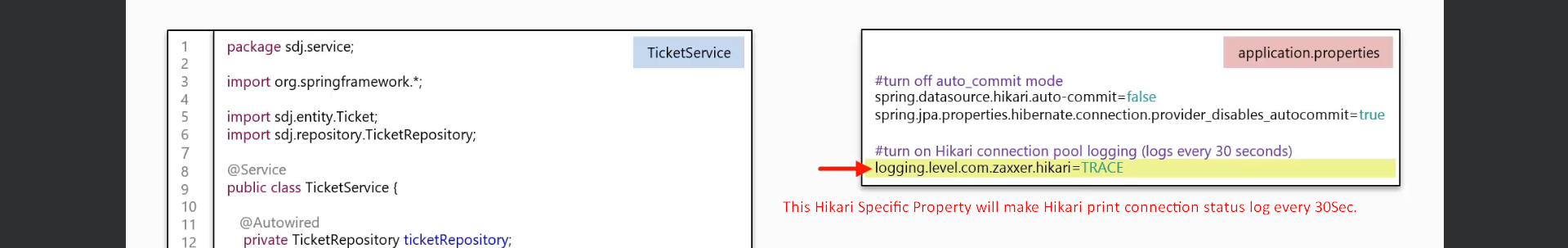
1. **Agenda**:
   1. In Spring Boot App, how to delay the DB Connection acquisition until it is really needed.
2. Before that, let’s spend one minute on connection pooling.
3. 
4. When you start app with Connection Pool, at the time of startup, the pool will acquire some connections and will store in pool ready for use.
5. Now when a SQL request, app will acquire the free connection from the pool rather than creating a new connection and when done DB operation, the connection will not be closed and will be sent back into the pool to serve other requestions.  
   Thus saving the time taken for opening and closing.
6. By default, Spring Boot relies on Connection Pooling Library which is production level library for pooling your DB Connections (**HikariCP**).
7.   
   Why Connection will be acquired at the beginning of the method call annotated with **@Transactional**?  
   **Answer**: Actually, to define a transactional boundary by Transaction Manager, the Transaction Manager needs to confirm if auto-commit=false on the connection from the pool. If connection auto-commit=true, it will set it to be false.

**NOTE**: If auto-commit=true during the sql executions, each execution will be committed in its own boundary.

1. We can delay the acquisition of the connection by configuring the following two properties.   
   
2. 
3. 