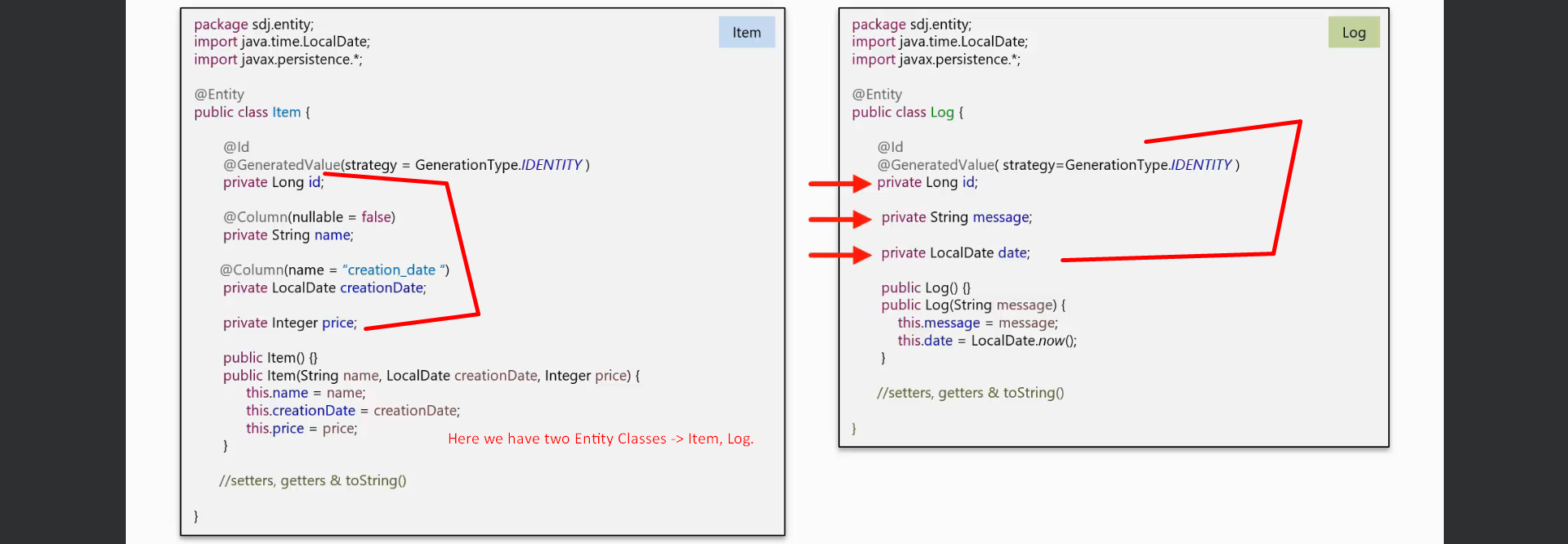
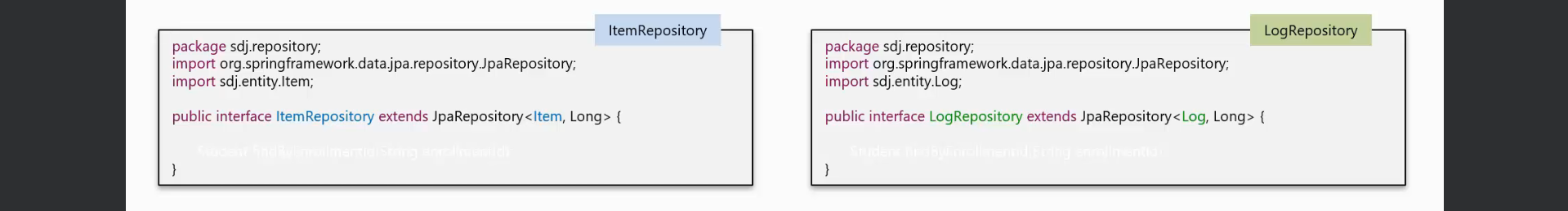
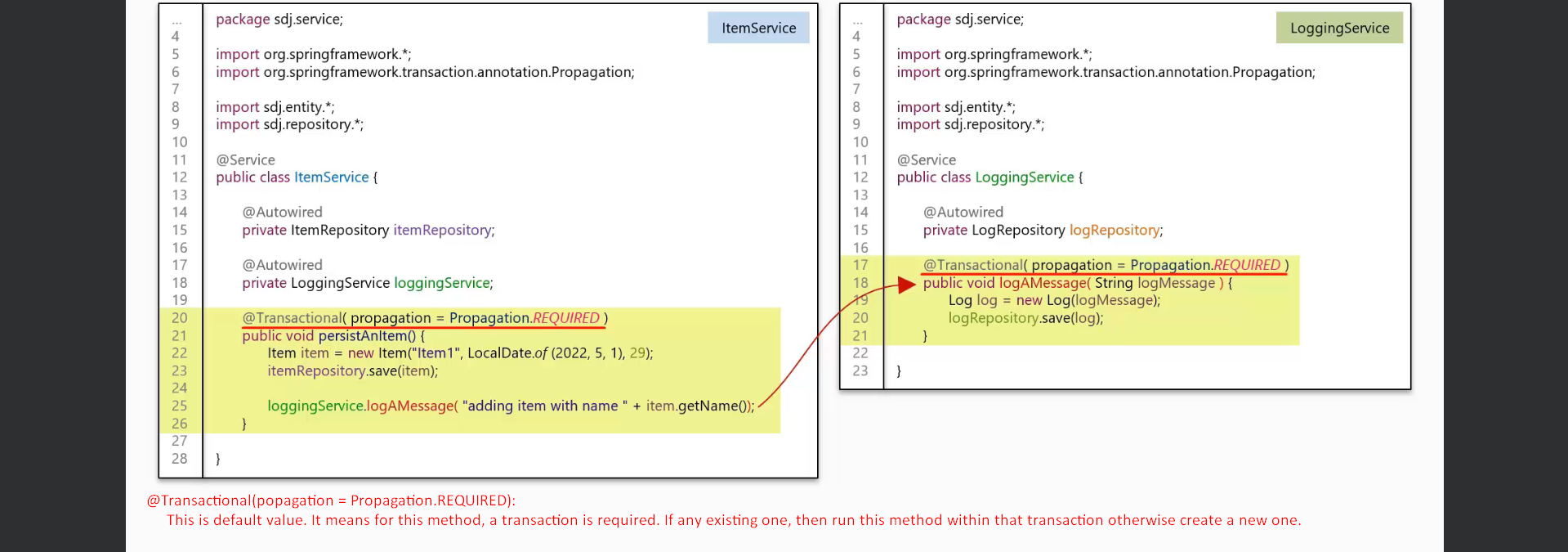
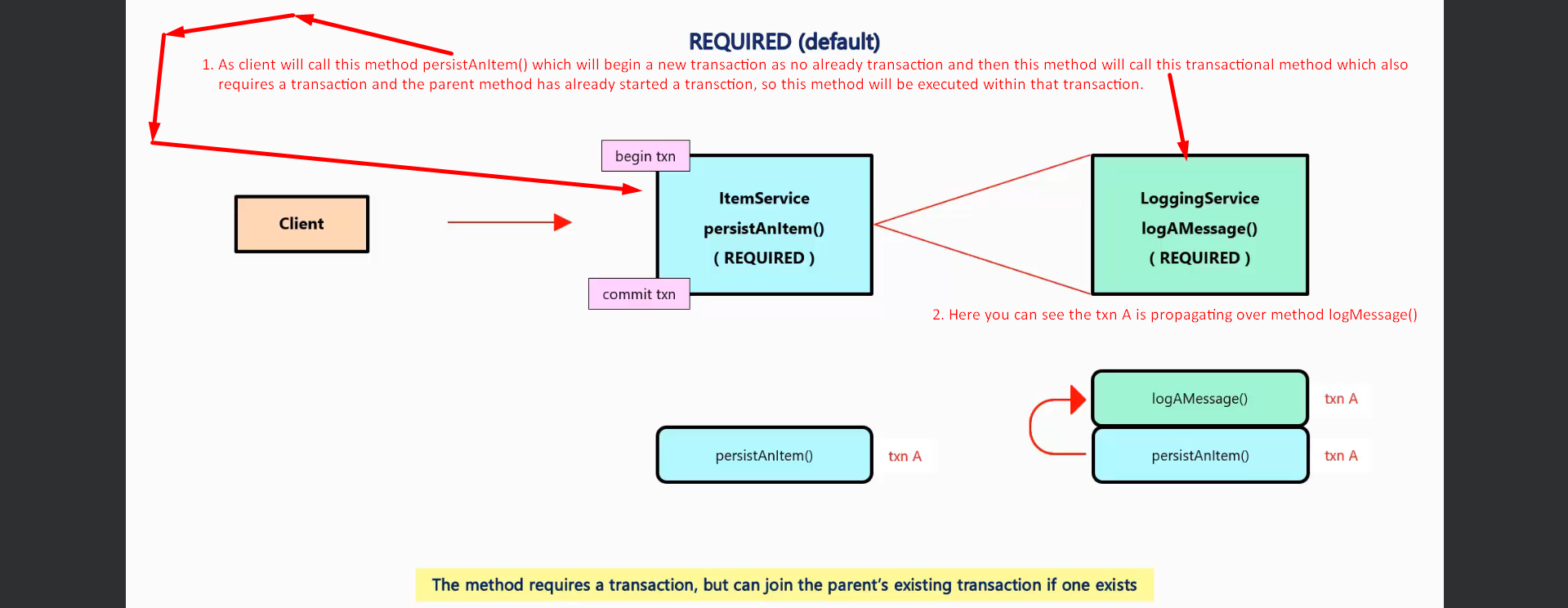
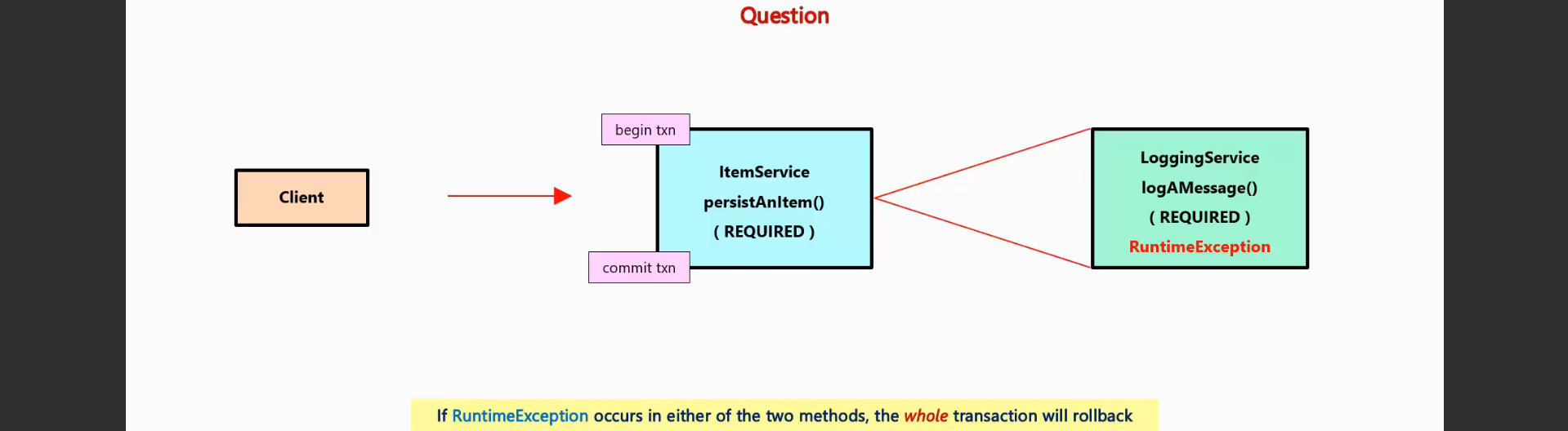
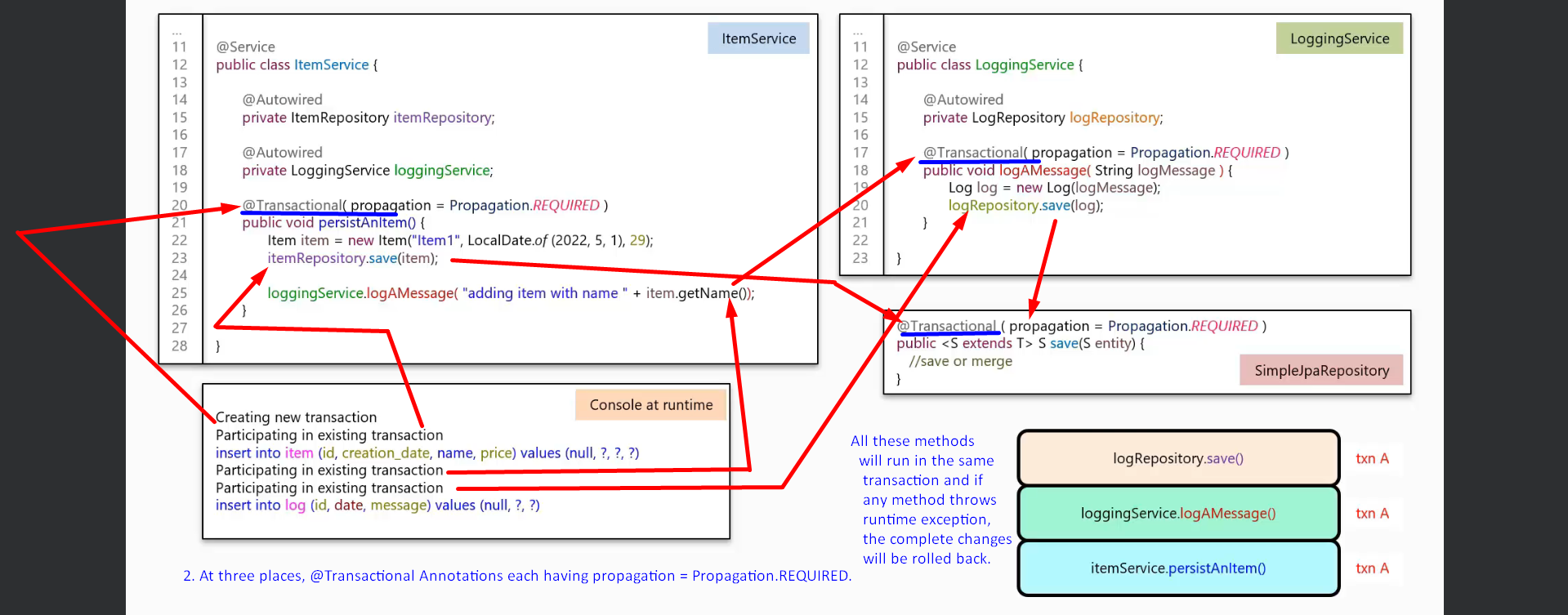
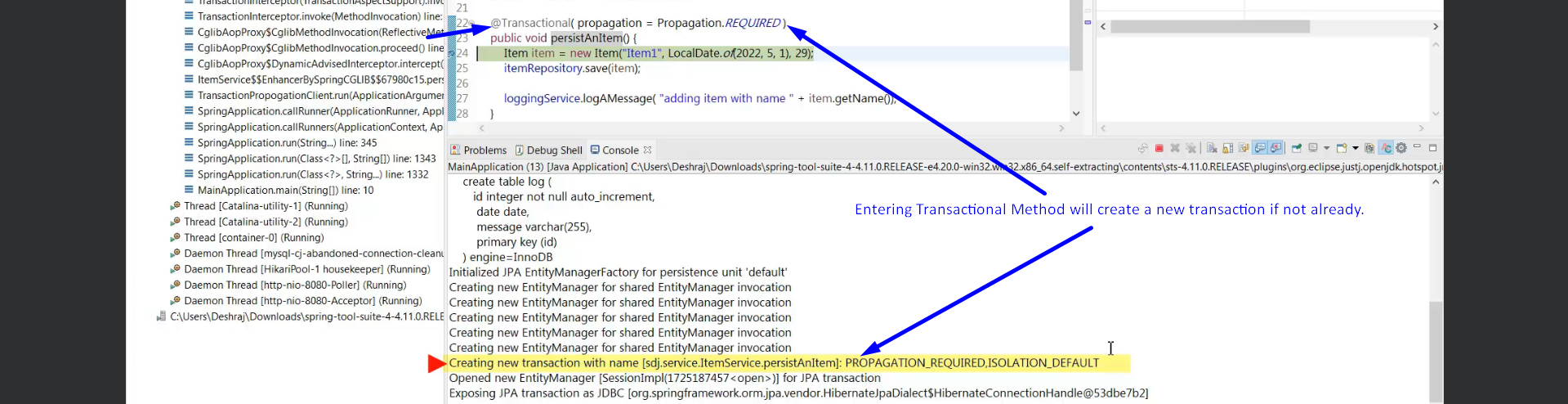
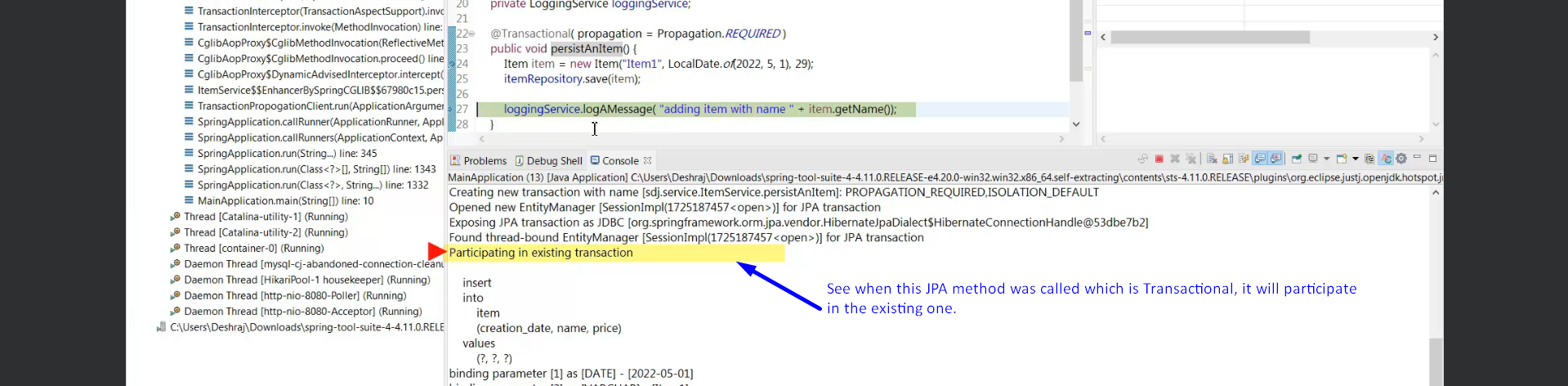
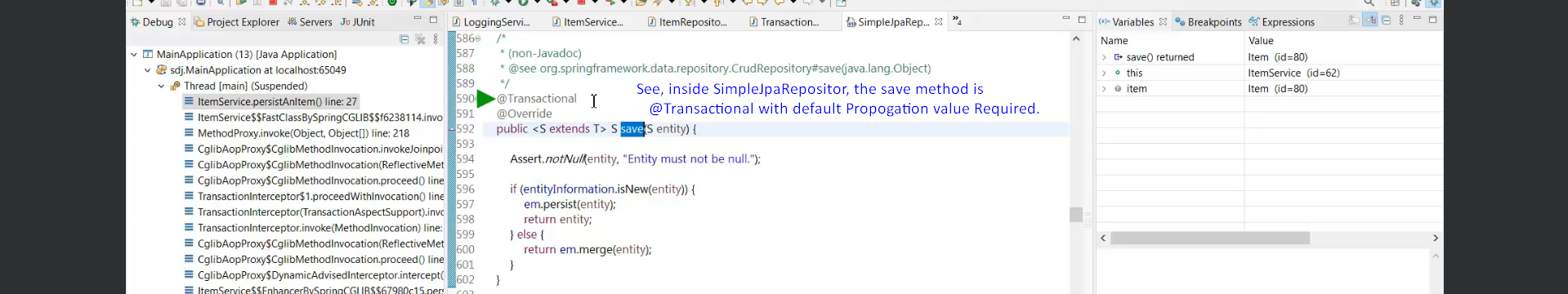
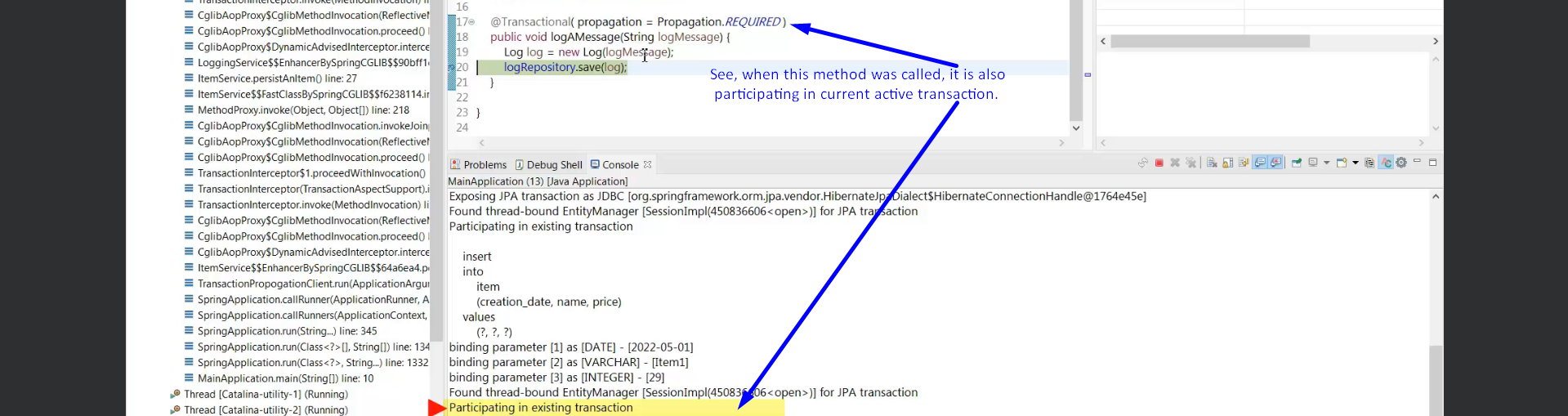
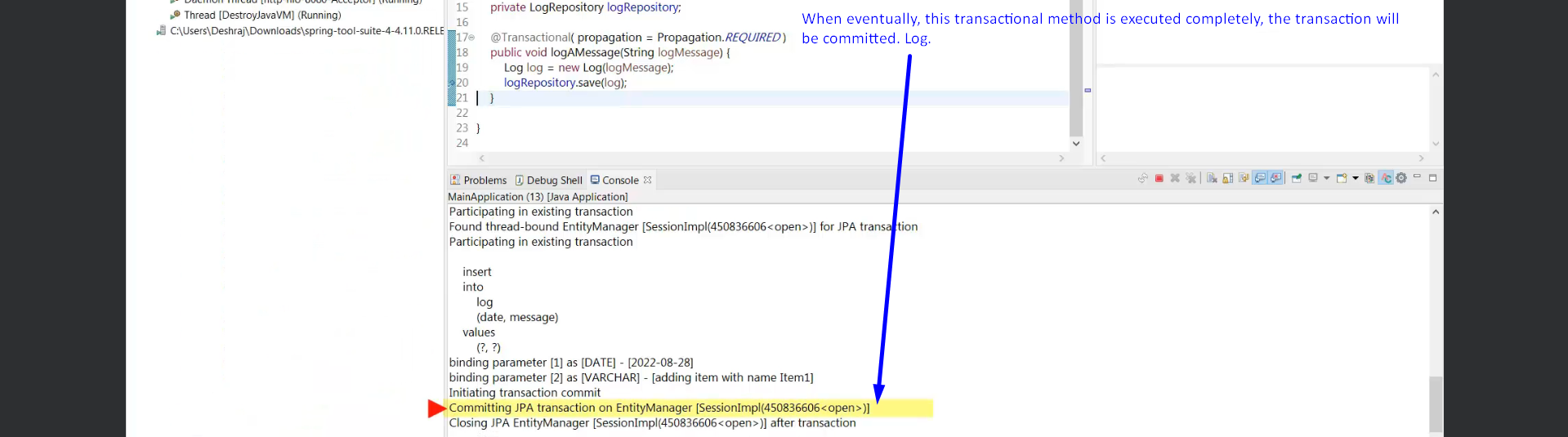
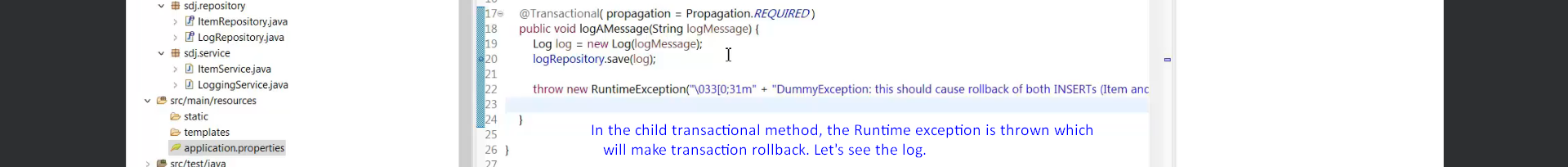
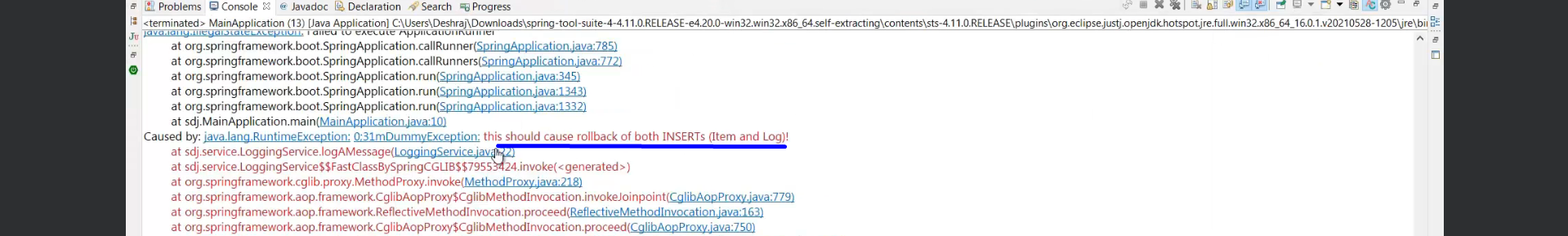
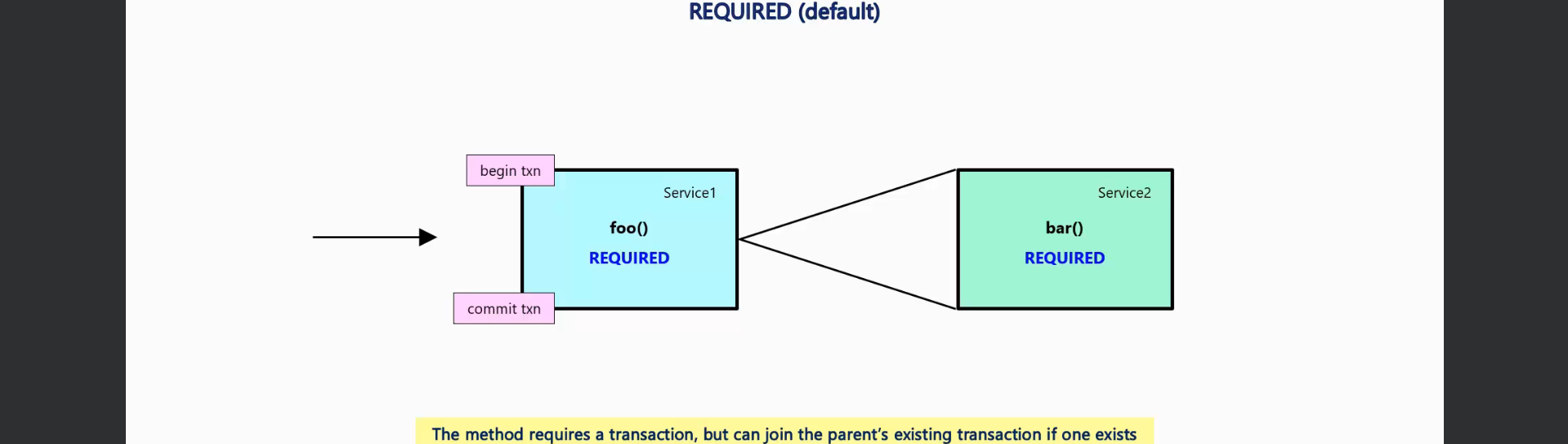
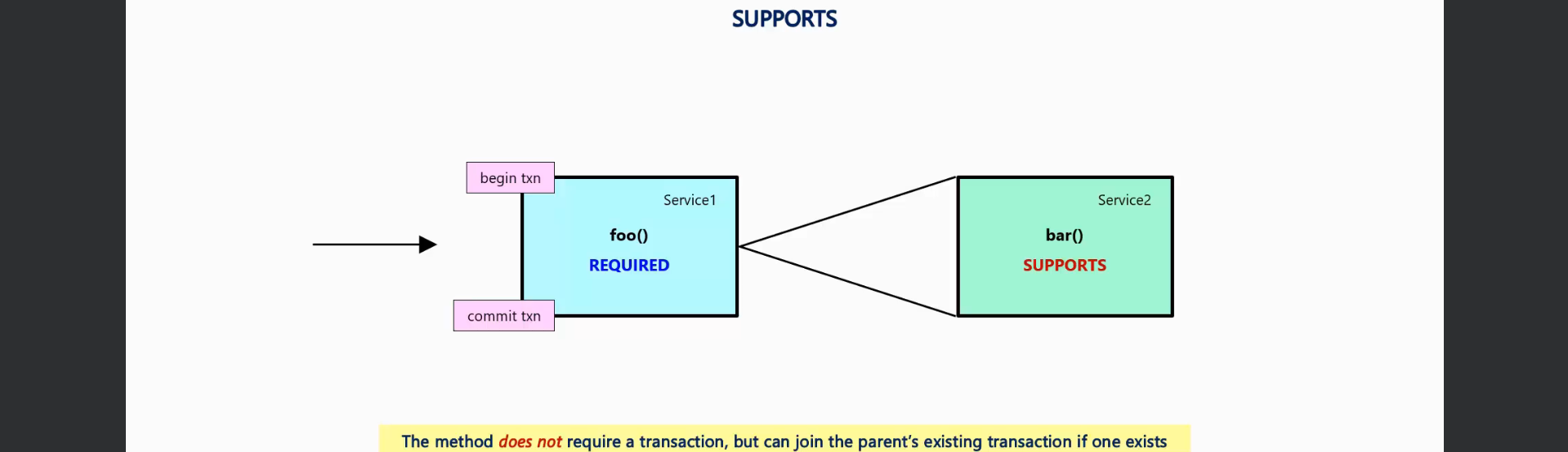
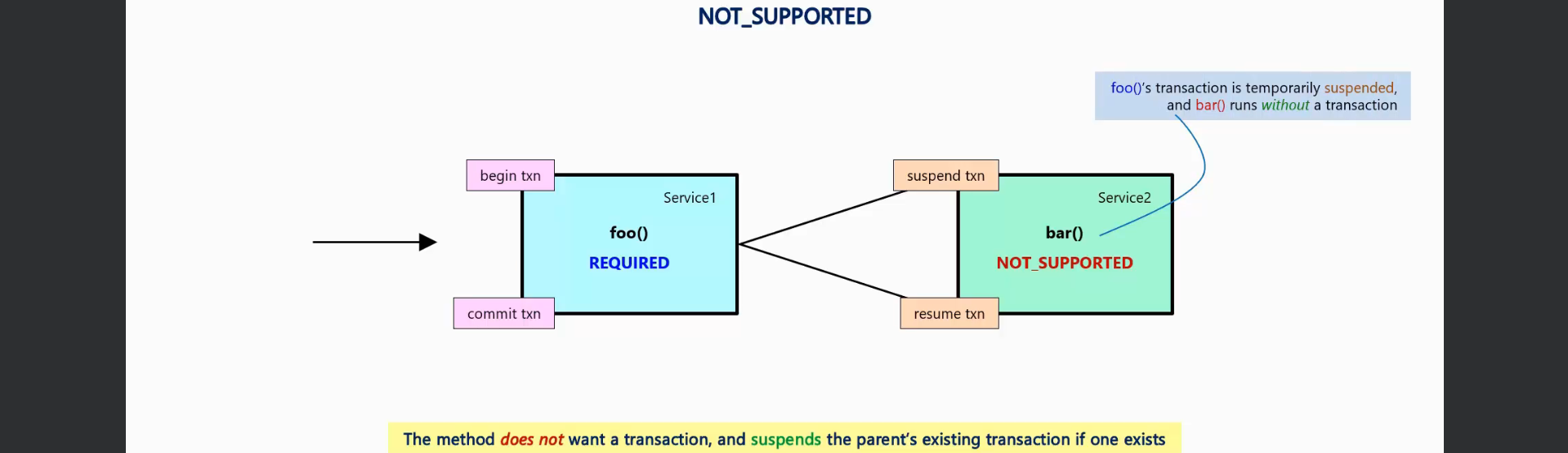
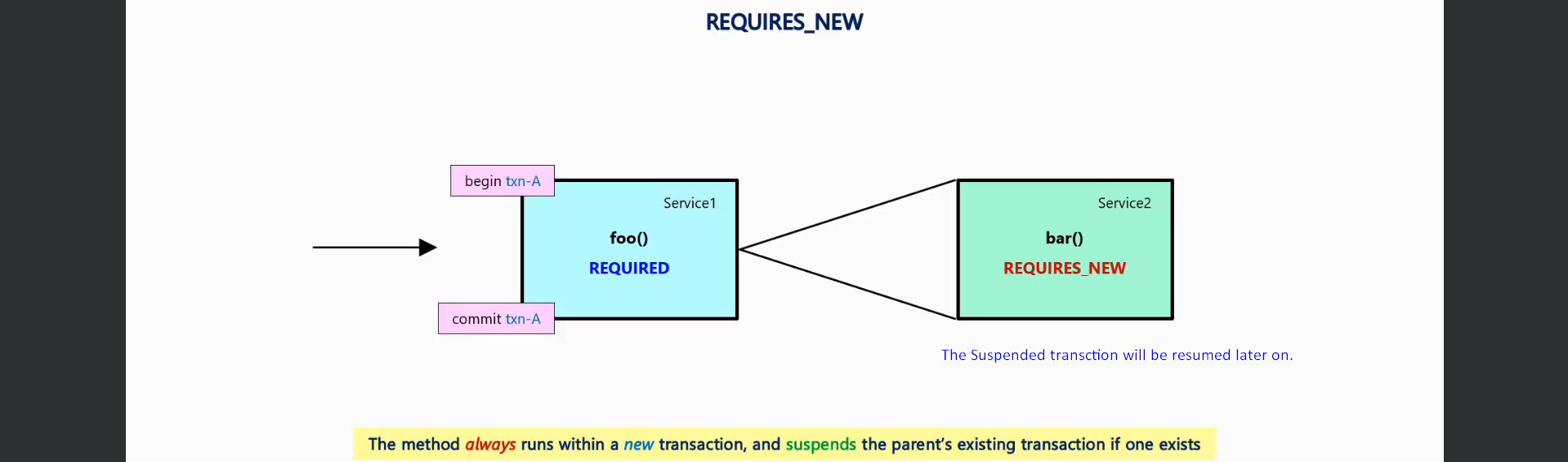
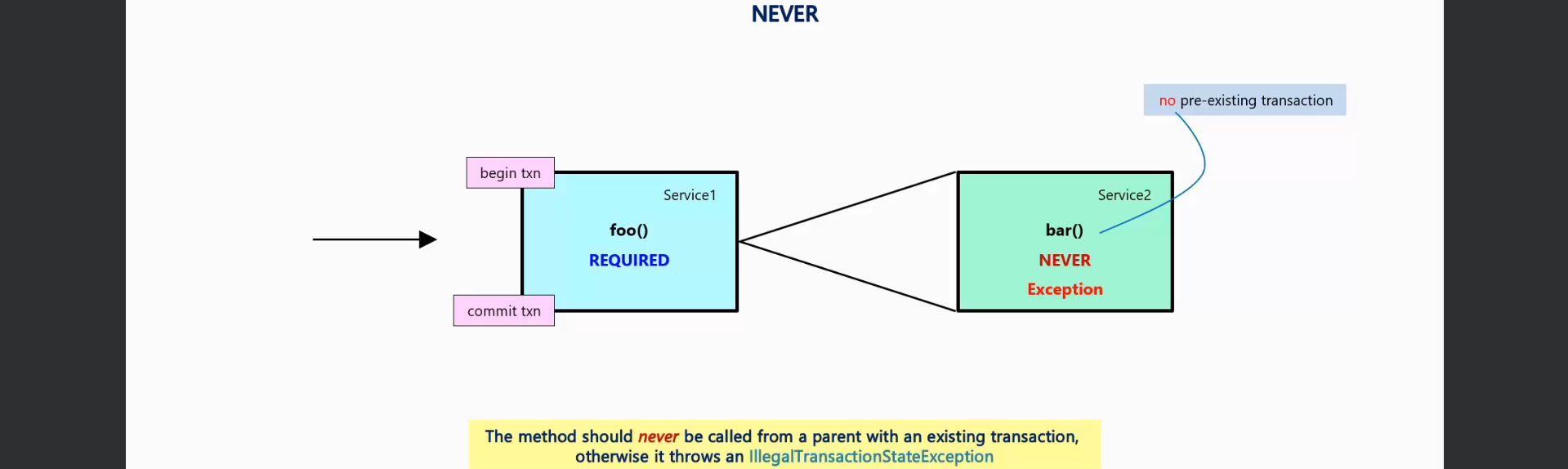
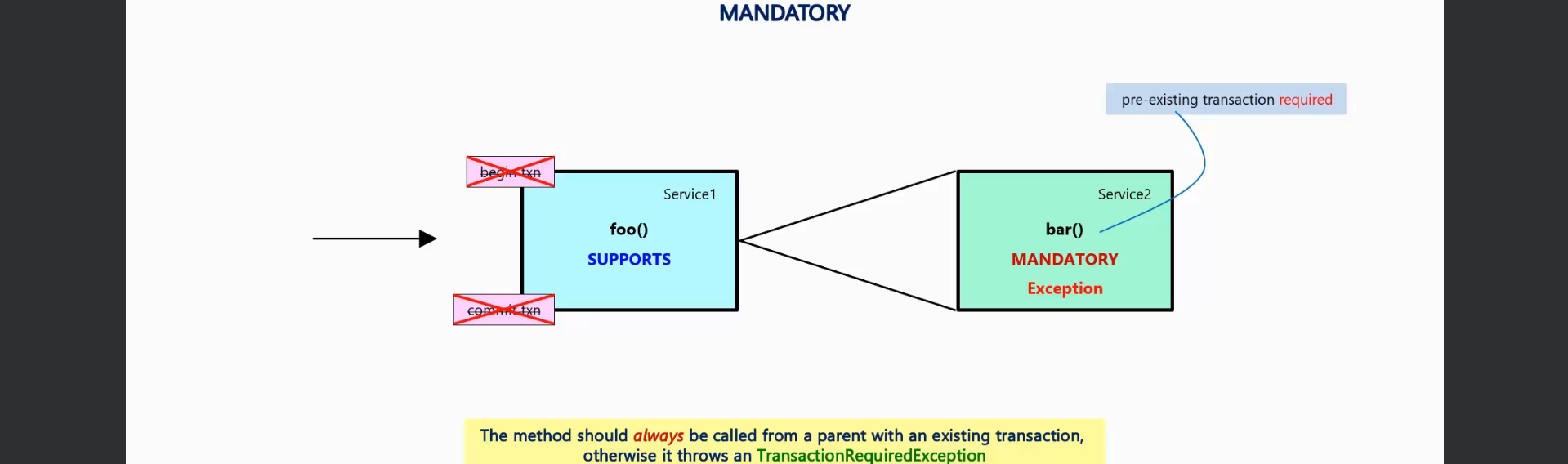
1. **Agenda**:
   1. Transaction Propagation: When one transactional method calls another transactional method.
2. Jatin From AI:   
   [Transaction](https://chatgpt.com/c/670529e3-6c30-8004-8d6f-63198aa5f9f0#:~:text=Transaction%20propagation%20in%20Spring%20defines%20how%20transactions%20interact%20with%20each%20other%20when%20multiple%20methods%2C%20each%20annotated%20with%20%40Transactional%2C%20are%20called%20in%20a%20sequence.%20It%20determines%20how%20a%20transactional%20method%20should%20behave%20when%20it%20is%20called%20by%20another%20method%20that%20is%20already%20part%20of%20a%20transaction) propagation in Spring defines how transactions interact with each other when multiple methods, each annotated with @Transactional, are called in a sequence. It determines how a transactional method should behave when it is called by another method that is already part of a transaction
3. The **propagation attribute** defines how the method should behave if a transaction already exists and whether the method should execute within a transaction context or not.
4. 
5. 
6. 
7. 
8. 
9. 
10. 
11. 
12. 
13.   
      
    
14. 
15. 
16.   
      
    
17. 
18. **PROPAGATION.REQUIRED**



1. **PROPAGATION.SUPPORTS**
2. **PROPAGATION.NOT\_SUPPORTED**  
   Later, the suspended transaction will be resumed.
3. **PROPOGATION.REQUIRES\_NEW**
4. **PROPAGATION.NEVER**  
   
5. **PROPAGATION.MANDATORY**  
   
6. **PROPAGATION.NESTED**  
   