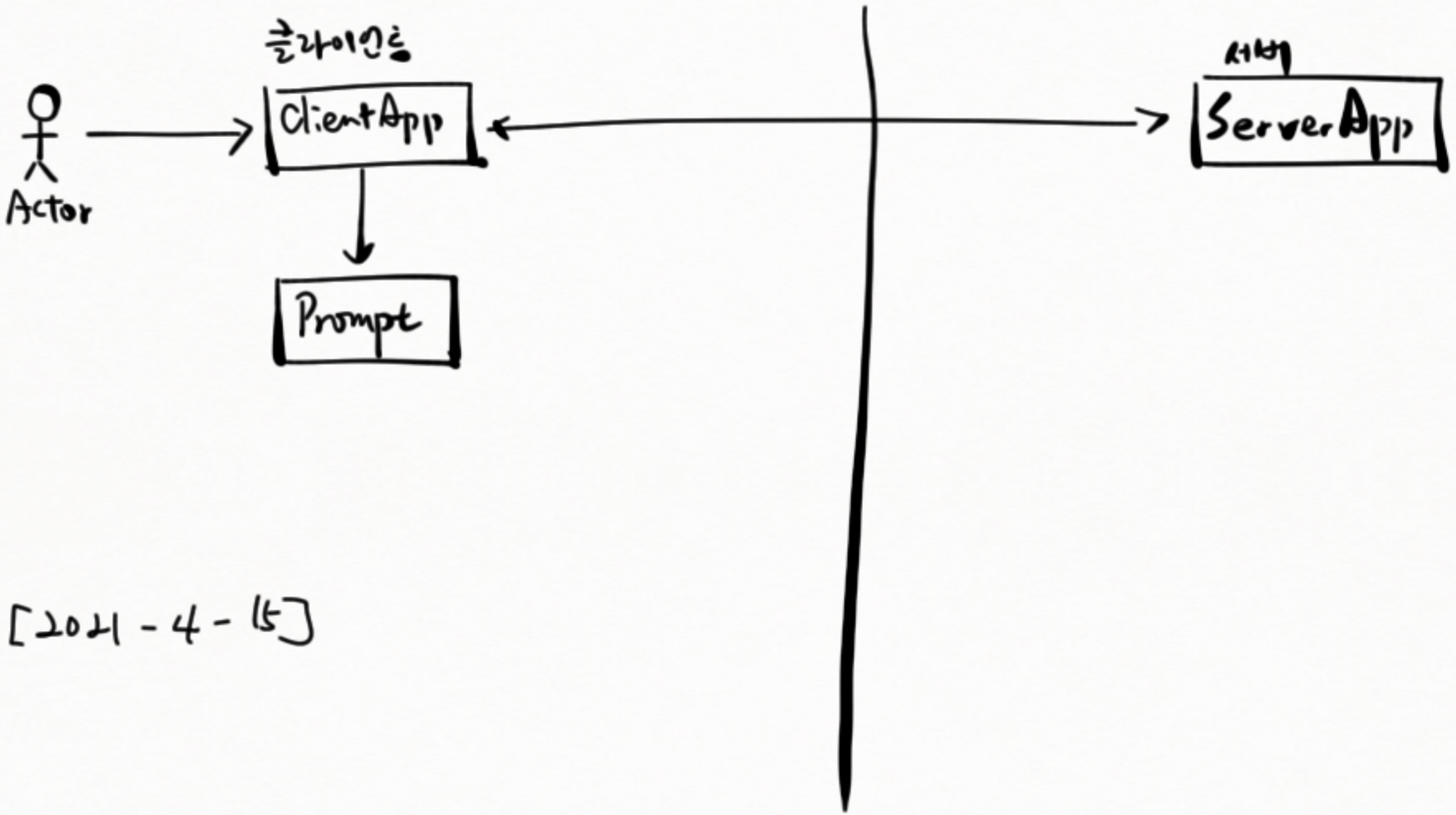
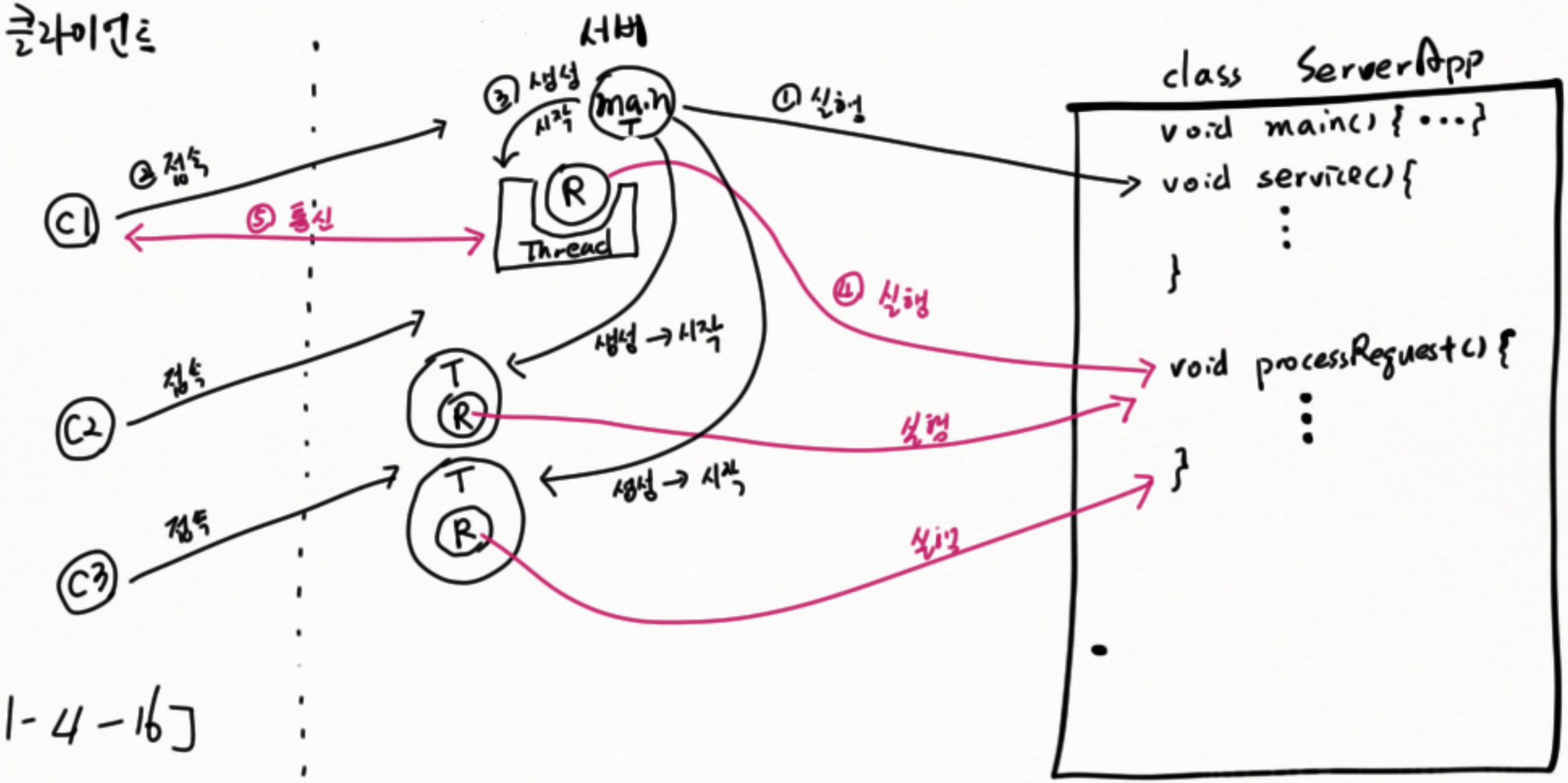
* 애플리케이션 서버 도입 - 1단계



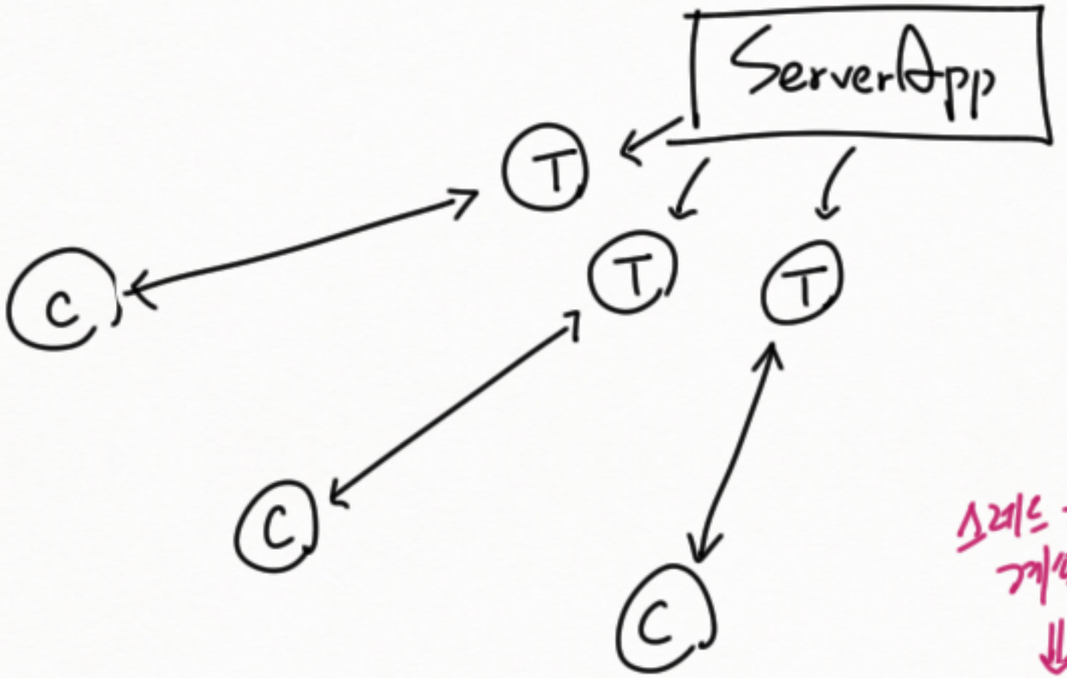
[2021-4-15]

* ServerApp의 쓰레드 구동 원리



[2021-4-16]

* Thread Pool 의 필요성



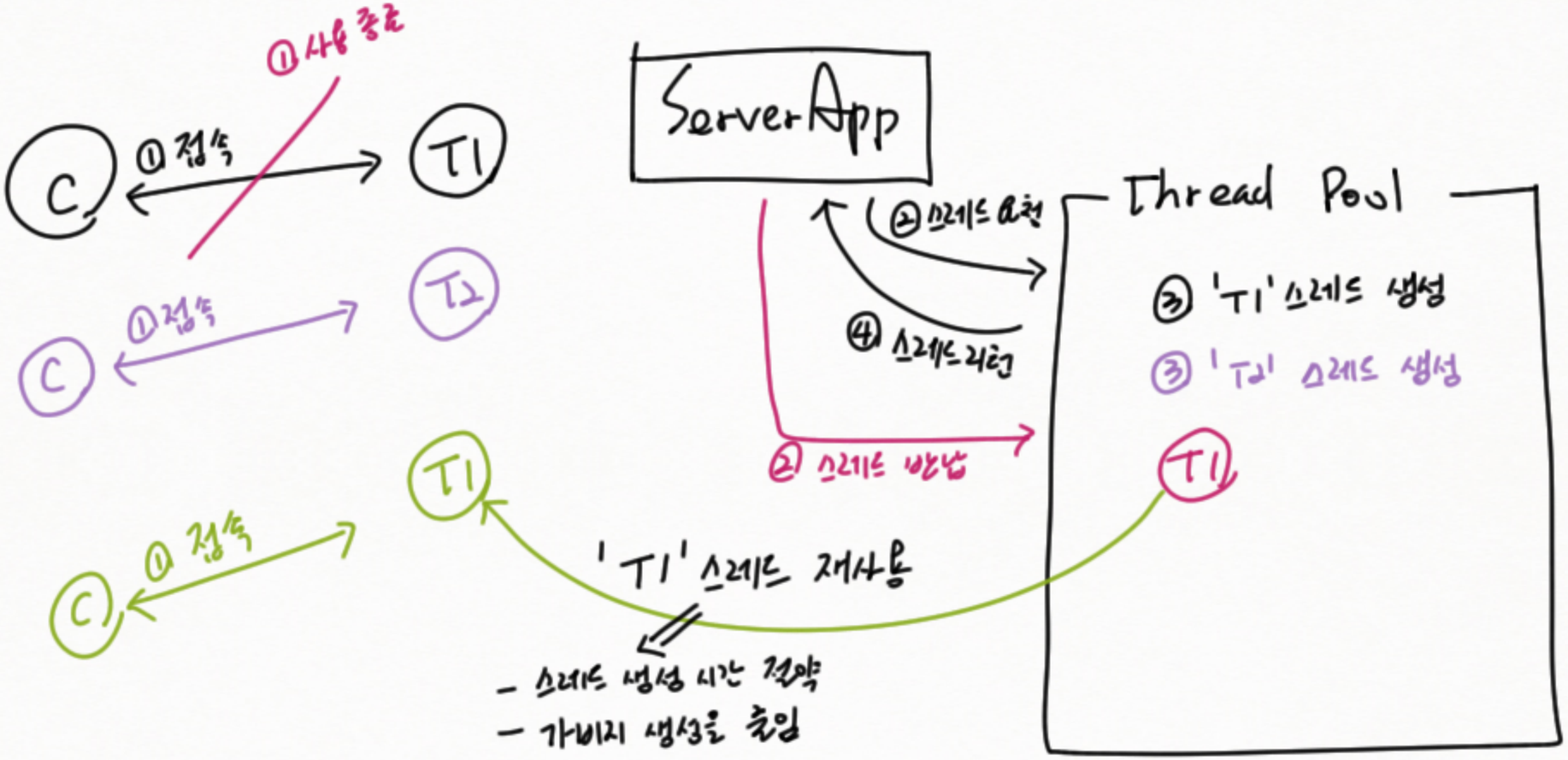
* 클라이언트가 접속할 때 마다
↓
스레드 생성
↓
실행
↓
소멸

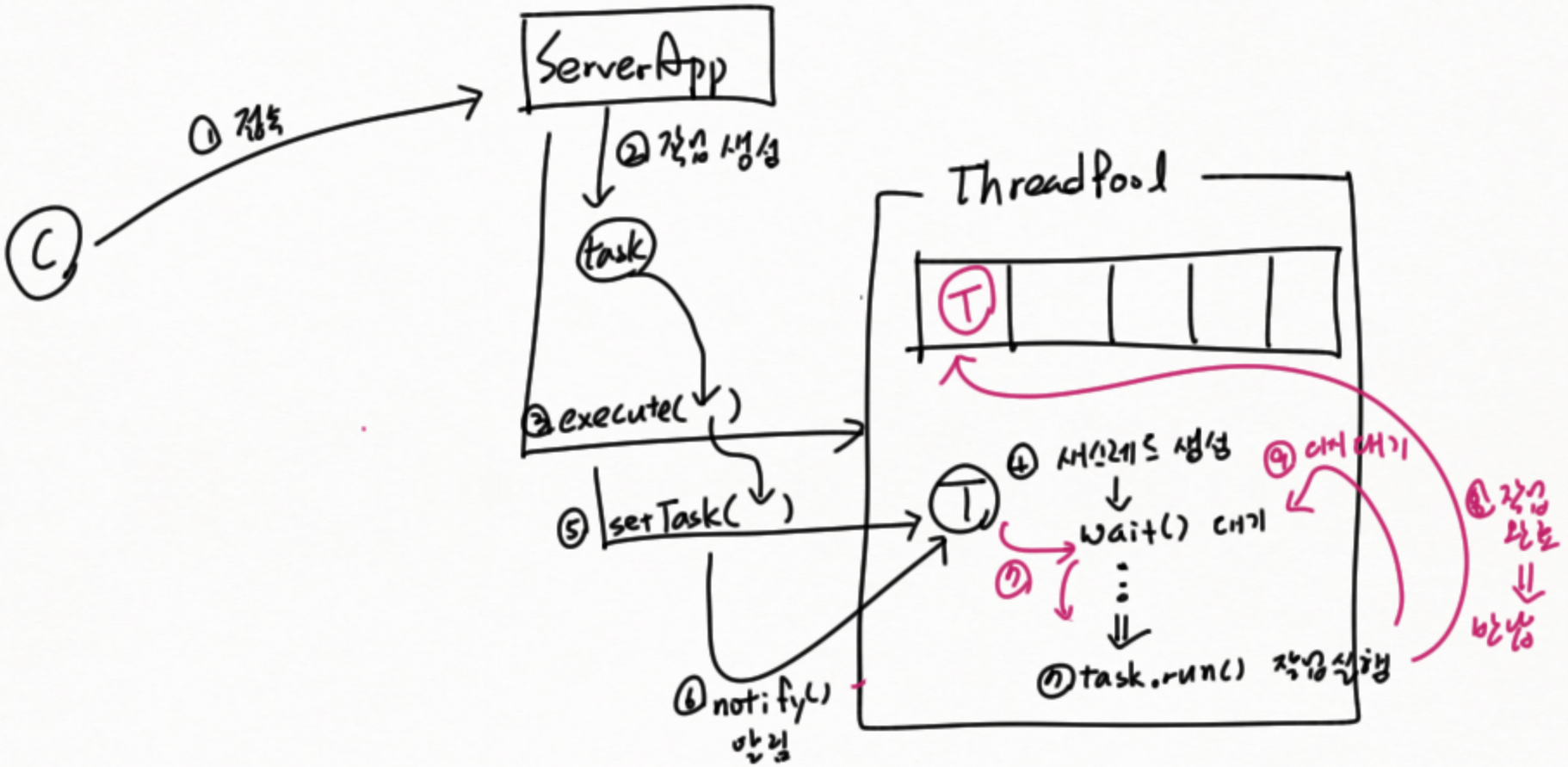
스레드 가비지가 계속 생성됨
⇓ 문제점

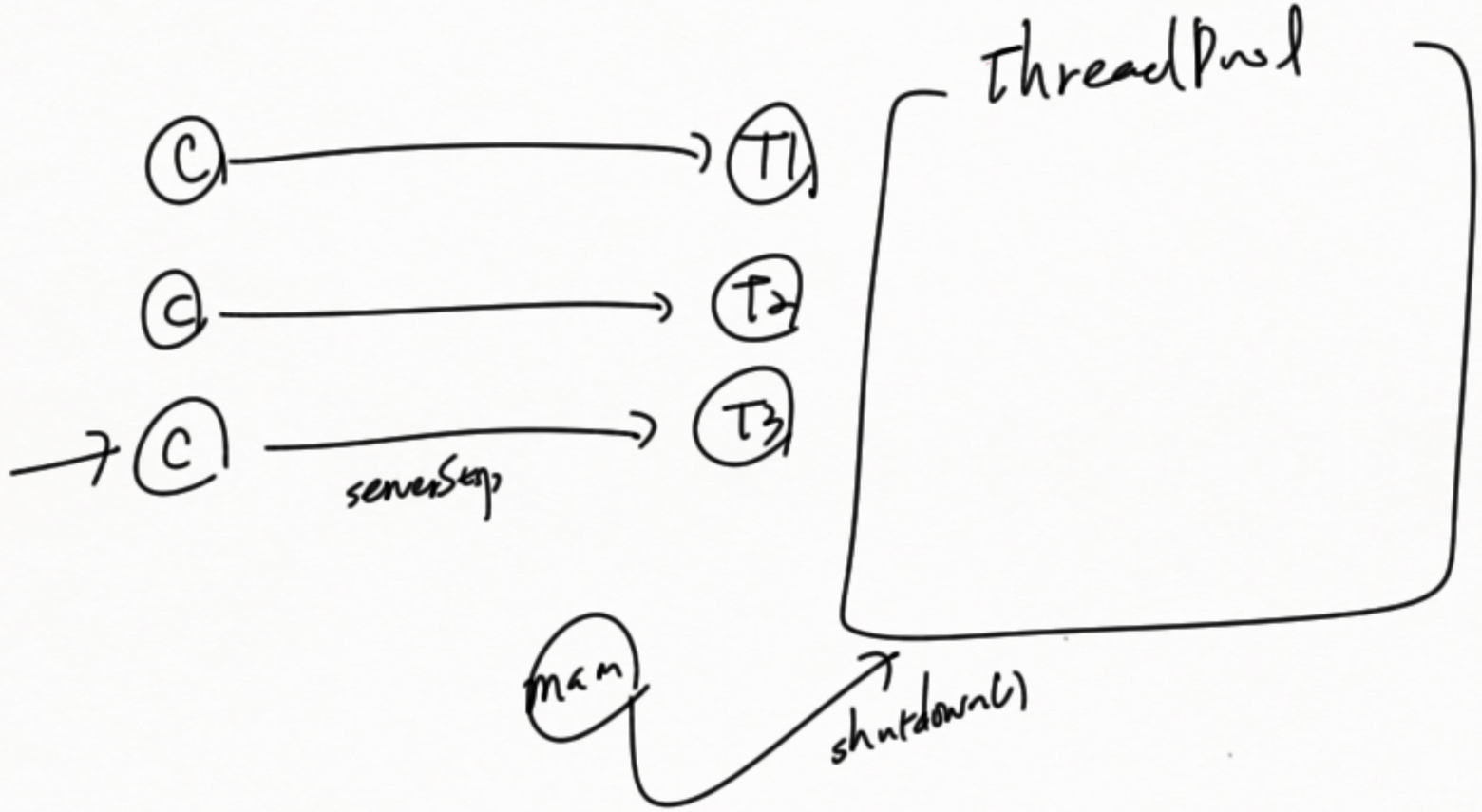
해결책? ← "스레드 재사용"
- 메모리 사용의 비효율성.
- 스레드 생성과 가비지 제거에 실행 시간 소요.

Pooling 기법으로 구현 ∈ "스레드 재사용"

* Thread Pool 구동 원리



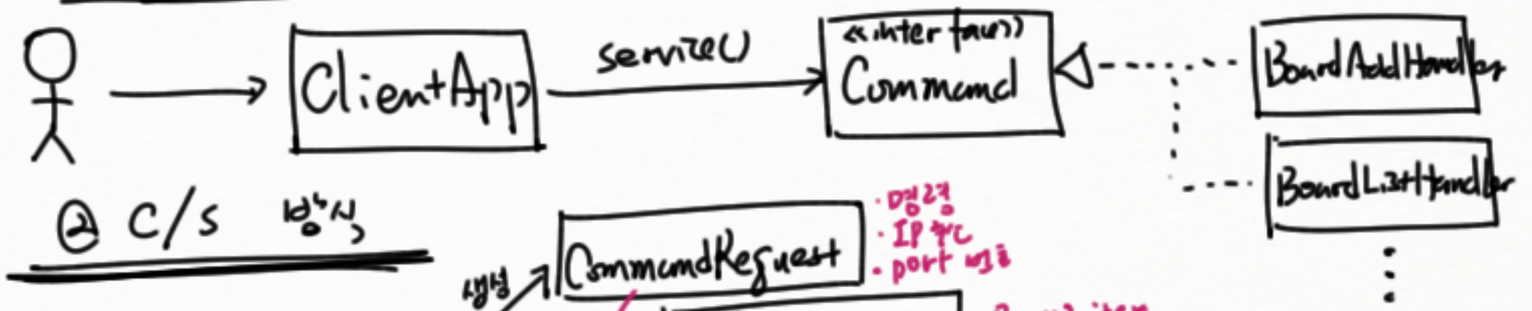
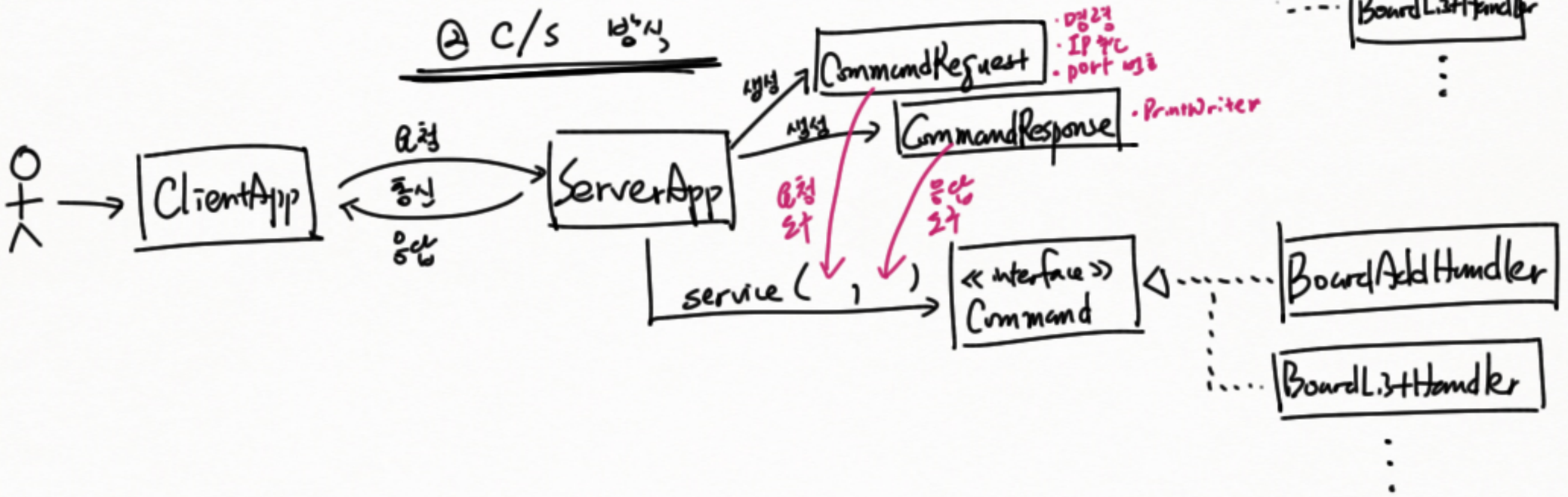

C
① 접속
ServerApp
② 작업 생성
task
③ execute( )
⑤ setTask( )
⑥ notify()
알림
ThreadPool
T
T
④ 새쓰레드 생성
wait() 대기
⑨ 계속 대기
⑦
⑦ task.run() 작업실행
⑧ 작업 완료
⇓
반납


ThreadPool
C
C
C
T1
T2
T3
serveStep
main
shutdown()

* Application 서버에서 Command 객체 실행

[2021-4-19]

① standalone 방식



② C/S 방식