

Virtual Thread

Java

Virtual Thread

CONTENTS

- 01** Platform Thread
- 02** Reactive Programming
- 03** Virtual Thread란?
- 04** Virtual Thread의 장단점
- 05** 예시 코드

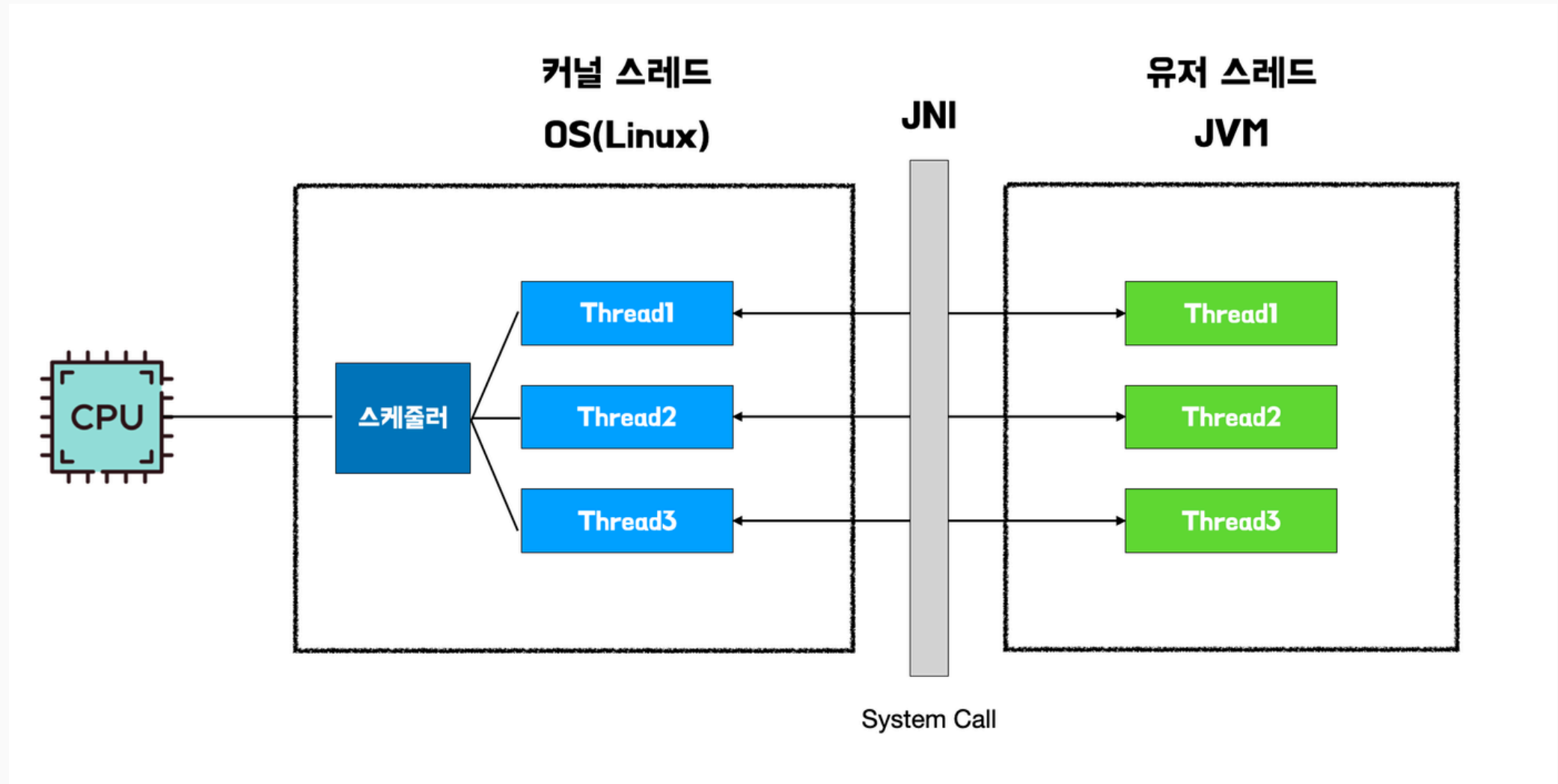
01

Platform Thread

Platform Thread

전통적인 Java의 Thread

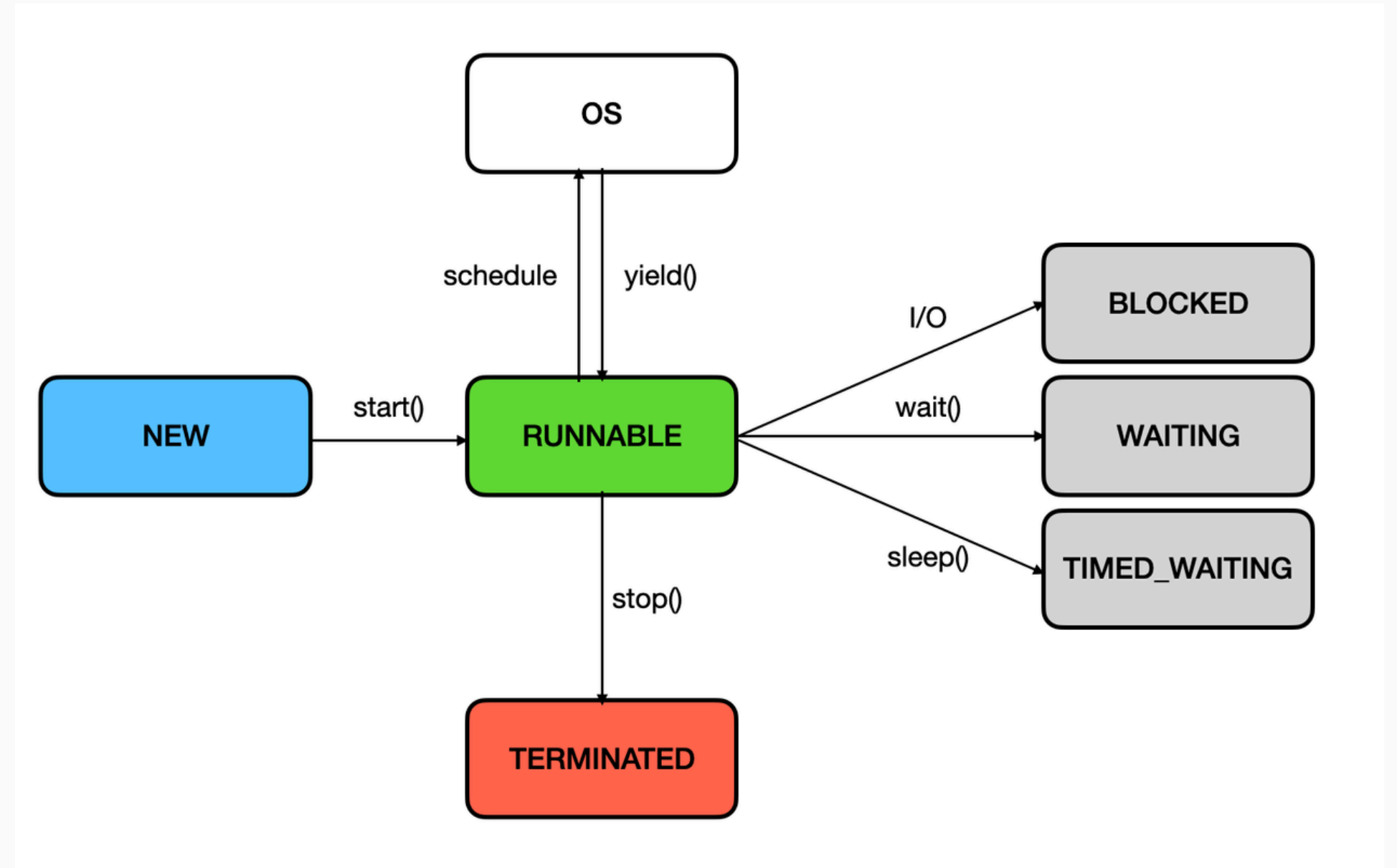
- 현재 Java의 Thread는 OS Thread와 1:1로 매칭됨
- Thread per Request
- Thread 개수 제한으로 인해 처리량이 제한되는 병목 현상 발생



Platform Thread

Context Switching

- 하나의 CPU에서 여러 Thread를 실행할 때, OS는 현재 Thread의 상태를 저장하고 다음 Thread의 상태를 불러옴
- 성능 저하의 원인



Platform Thread

Blocking I/O

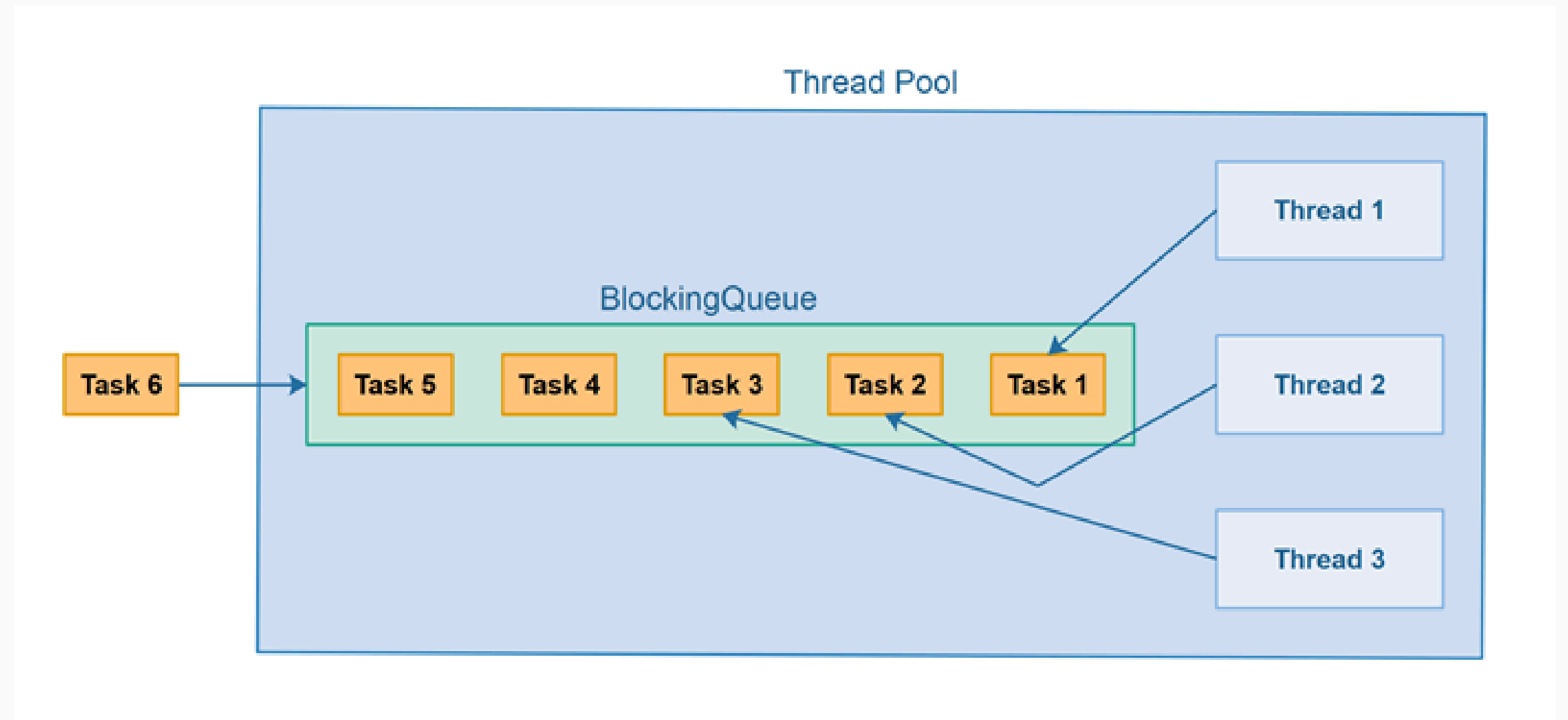


- Thread가 I/O 작업을 기다리는 동안 OS Thread 자원을 계속 점유함
- 작업을 처리하는 시간보다 대기하는 시간이 긴 경우?

Platform Thread

Thread Pool

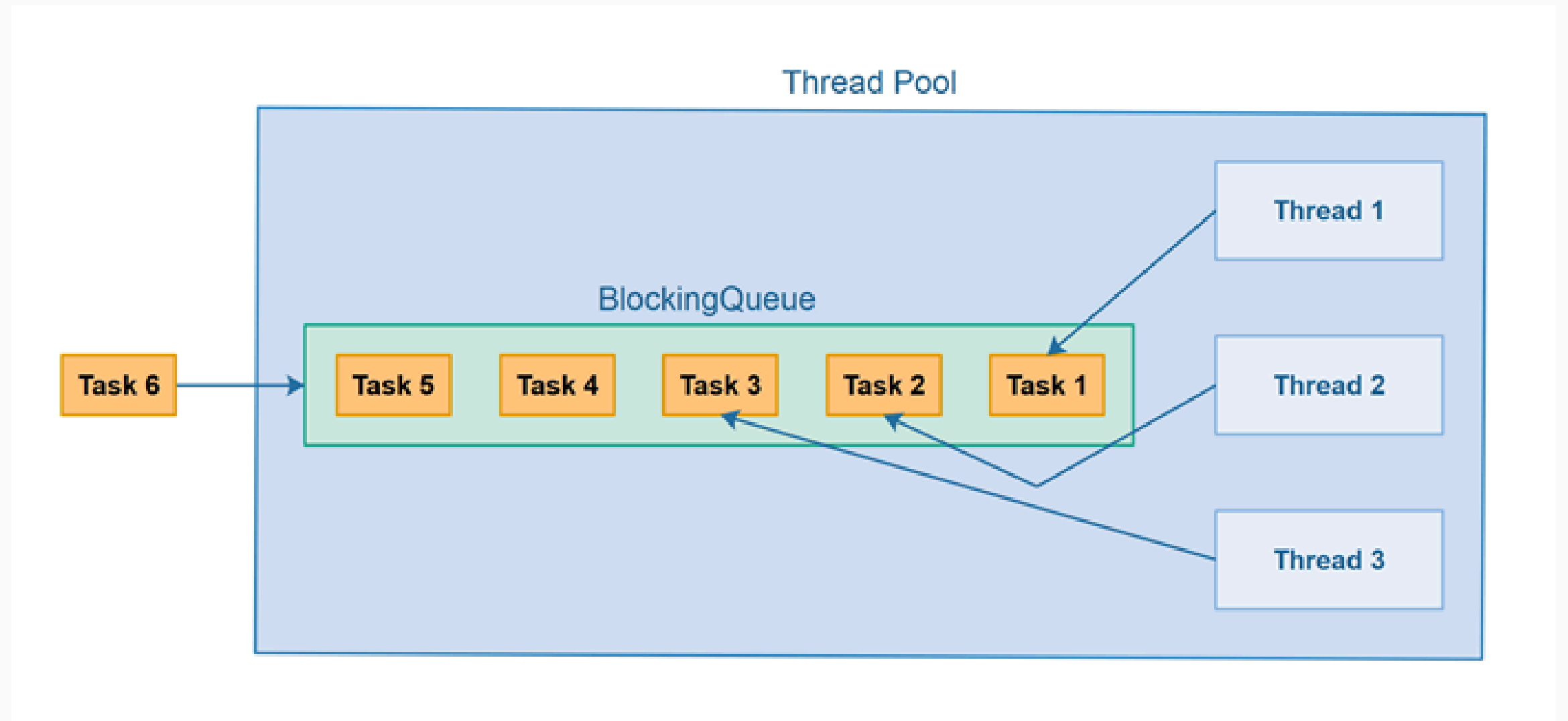
- 미리 만들어 둔 여러 개의 Thread를 재사용하여 작업을 효율적으로 처리하는 구조



Platform Thread

Thread Pool

- Queue에 있는 작업을 일이 없는 Thread가 맡아서 처리
- 처리가 끝나면 다시 작업 대기
- 제한된 개수의 Thread 운용

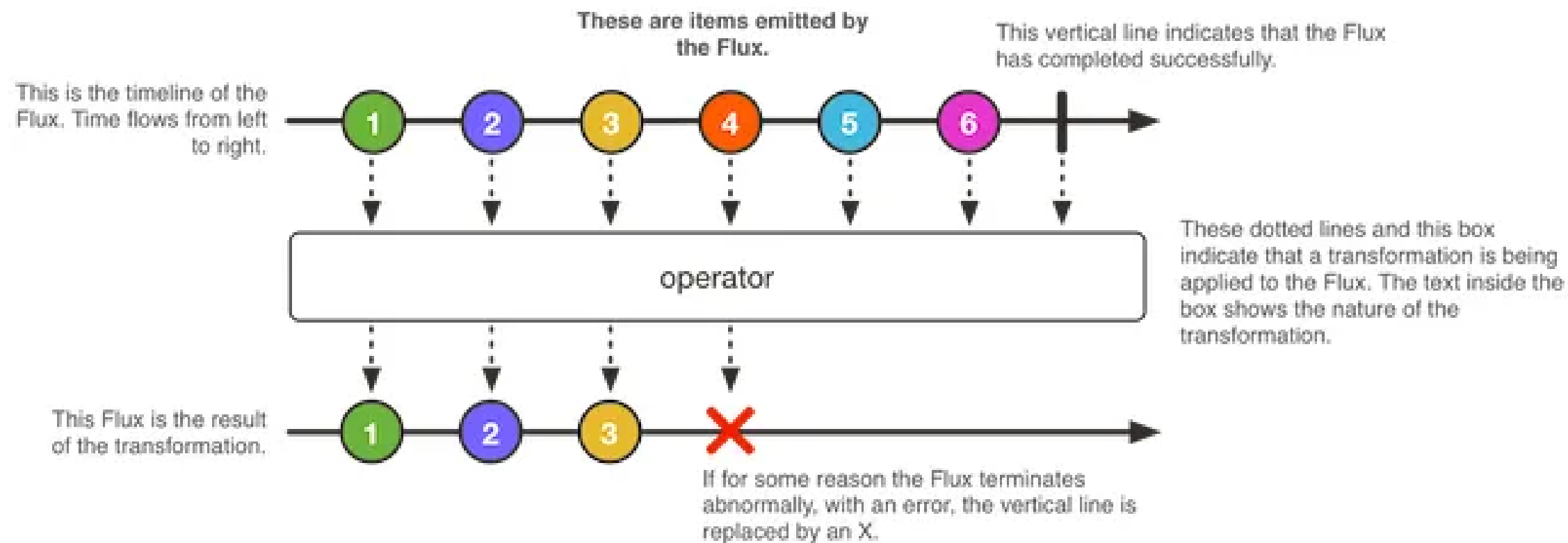


02

Reactive Programming

Reactive Programming

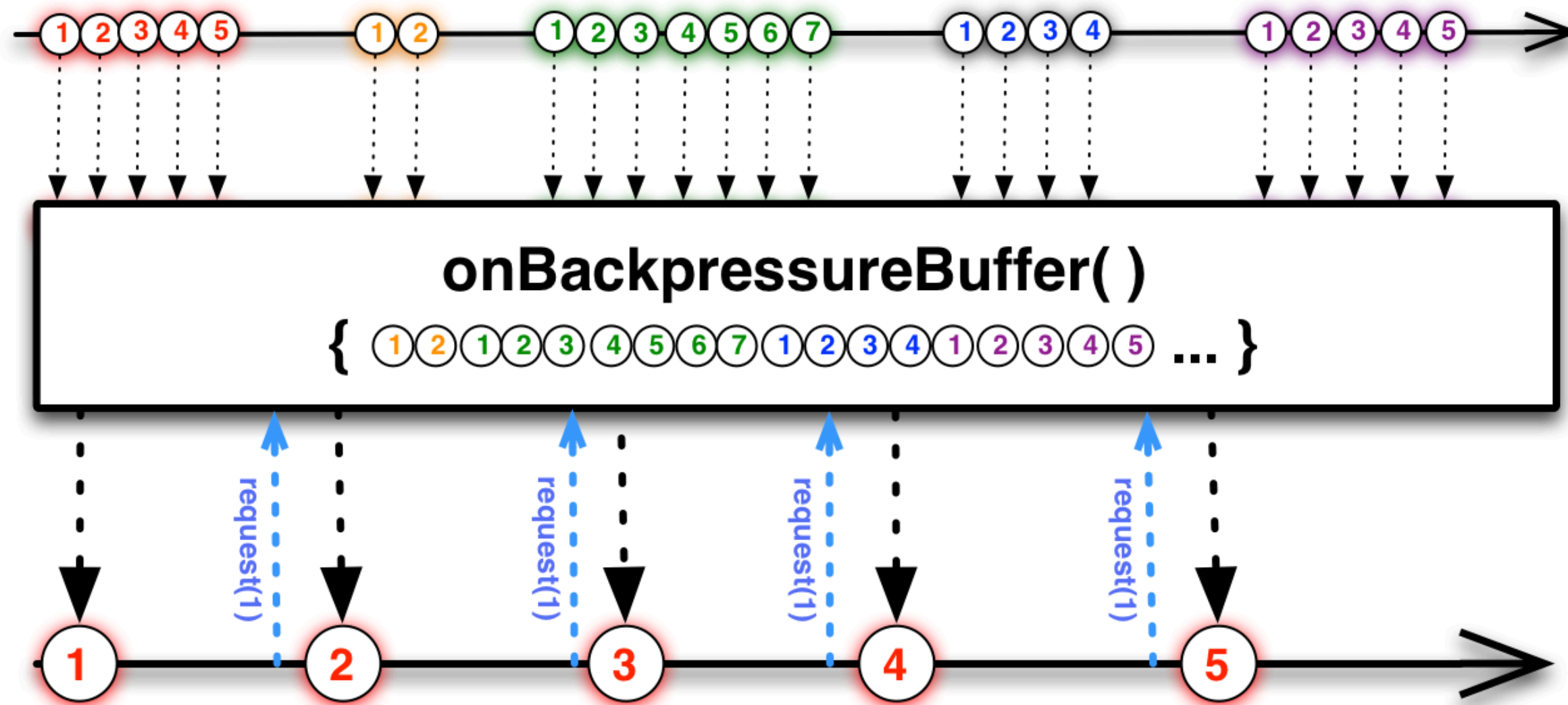
- 데이터의 흐름과 변화의 전파에 중점을 둔 비동기 프로그래밍 패러다임
- non-blocking 방식으로 동작



Reactive Programming

코드 이해의 어려움

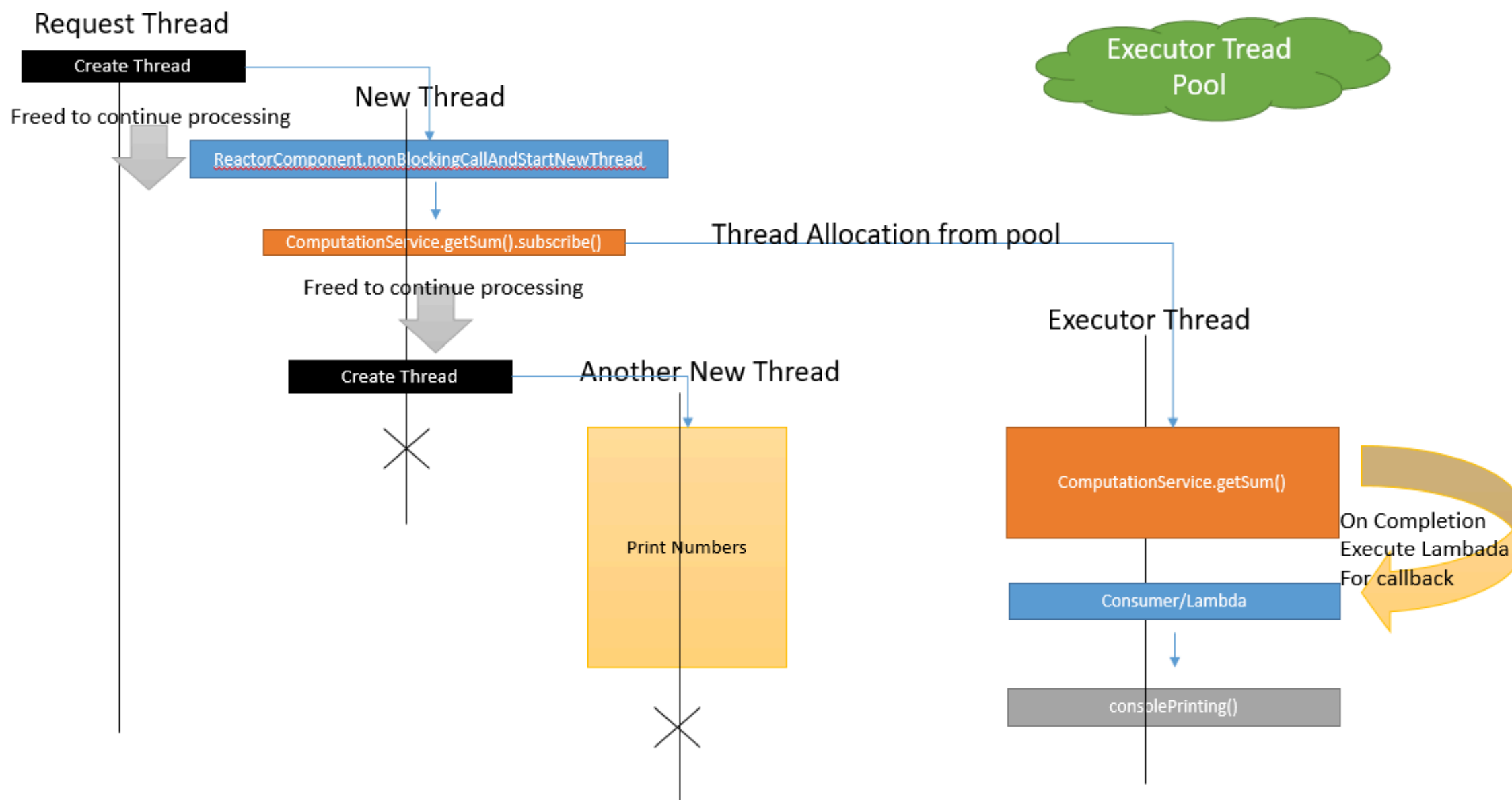
- 새로운 기능을 추가한다면 어디에 추가해야 할까?



Reactive Programming

디버깅과 예외 처리

- 기존에는 콜스택을 추적하면 디버깅과 예외 처리가 가능
- Request를 여러 Thread에서 처리하게 되면서 디버깅과 예외 처리가 힘들어짐

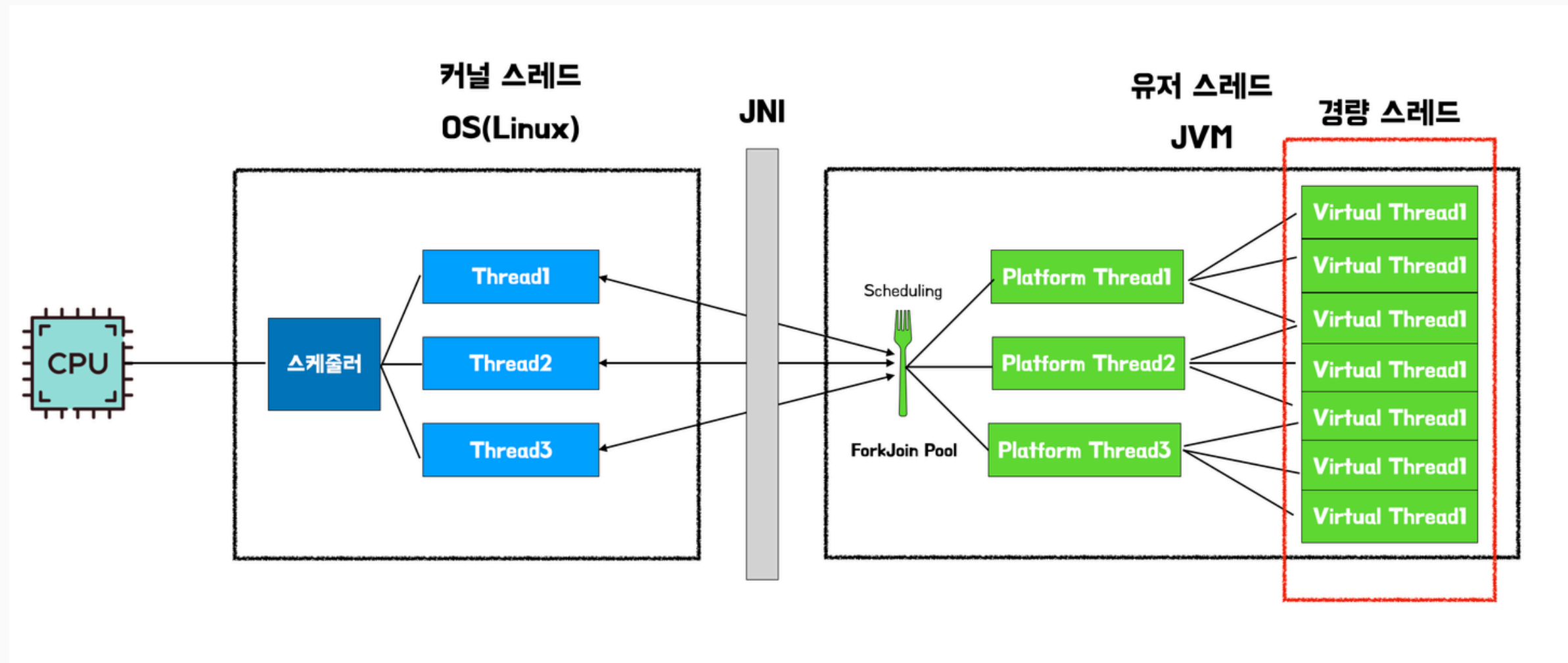


03

Vitruual Thread

Virtual Thread

- JVM이 직접 관리하는 경량 Thread 모델
- Blocking I/O 작업 시 다른 Virtual Thread 실행 가능



Virtual Thread

JEP 425: Virtual Threads (Preview)

Authors Ron Pressler, Alan Bateman
Owner Alan Bateman
Type Feature
Scope SE
Status Closed / Delivered
Release 19
Component core-libs
Discussion loom dash dev at openjdk dot java dot net
Effort XL
Relates to [JEP 436: Virtual Threads \(Second Preview\)](#)
Reviewed by Alex Buckley, Brian Goetz, Chris Hegarty
Created 2021/11/15 16:43
Updated 2023/06/07 09:43
Issue 8277131

JEP 444: Virtual Threads

Author Ron Pressler & Alan Bateman
Owner Alan Bateman
Type Feature
Scope SE
Status Closed / Delivered
Release 21
Component core-libs
Discussion loom dash dev at openjdk dot org
Relates to [JEP 436: Virtual Threads \(Second Preview\)](#)
Reviewed by Alex Buckley
Endorsed by Brian Goetz
Created 2023/03/06 18:00
Updated 2024/07/10 14:29
Issue 8303683

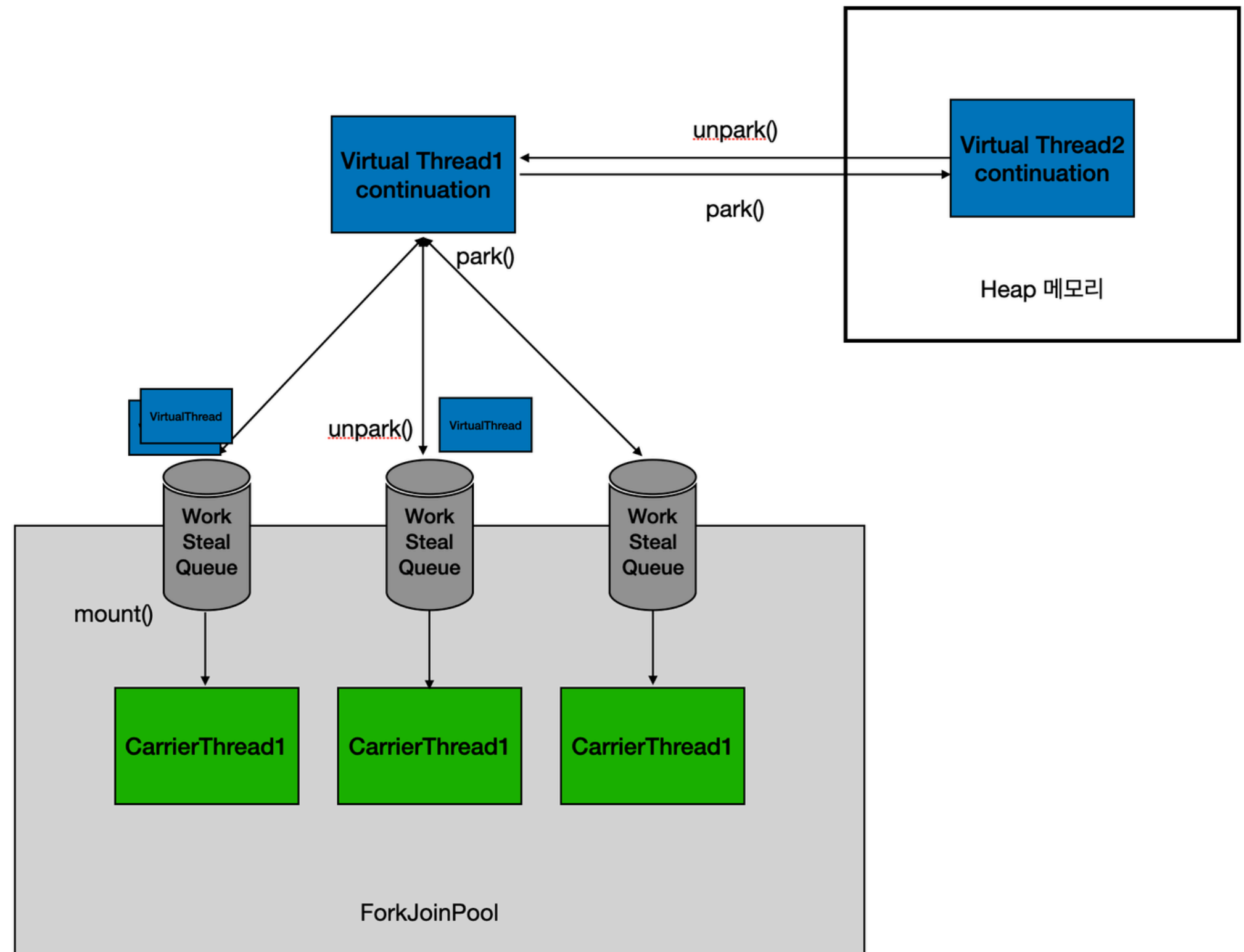
지원 버전

- JDK 19 얼리 액세스
- JDK 21 정식 도입

Virtual Thread

동작 흐름

- 대기
- mount()
- park()
- unpark()



Virtual Thread

구분	Platform Thread	Virtual Thread
Metadata size	약 2KB (OS별 차이 있음)	200~300B
Memory	미리 할당된 Stack 사용	필요시마다 Heap 사용
Context Switching cost	1~10 μ s (커널 영역에서 발생)	ns

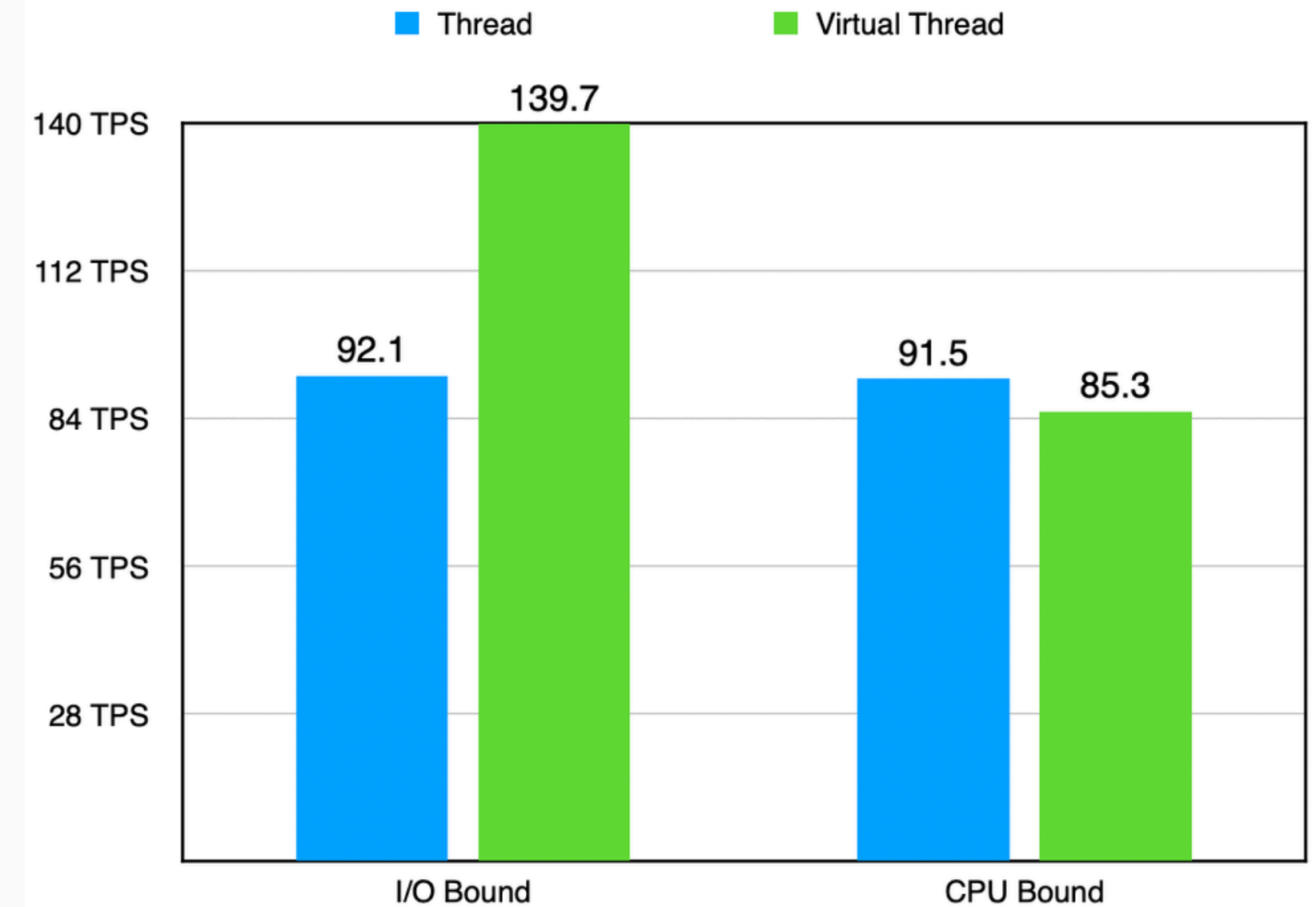
04

장점 및 단점

Platform Thread

장점

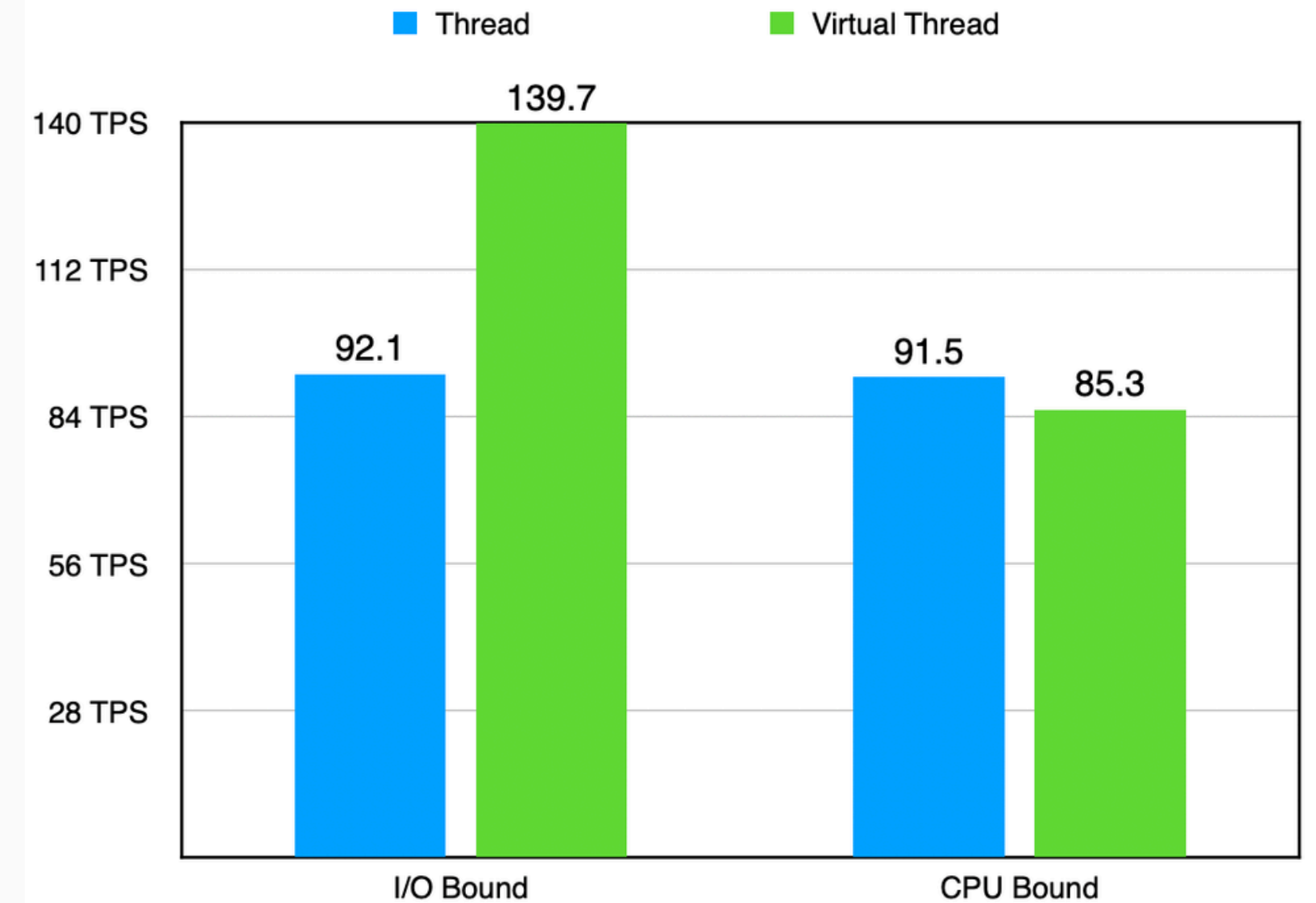
- 효율적인 Thread 생성
- 리소스를 낭비하지 않고 blocking 작업 처리 가능
- 동일한 하드웨어로 더 많은 요청 동시 처리 가능
- 기존 Java Thread 코드와 호환



Platform Thread

단점

- I/O 대기 시간이 긴 작업에는 적합하지만 CPU 집약적인 작업에는 부적합



Platform Thread

단점

- ‘ThreadLocal’ 변수 사용 시 고려
 - 메모리 누수와 성능 저하의 원인
 - Thread Pooling의 비효율성
 - 사용할 때 새롭게 인스턴스 생성 후 사용 후 버림
-